Trabajo de Fin de Grado

-

Especificación y diseño de un sistema software para la realización de auditorías informáticas

Autor: Eloi Roca Corcelles

Ponente: Joan Antoni Pastor (ESSI)

Grau en Enginyeria Informatica (GEI)

Enginyeria del Software

30 de Junio de 2020

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Índice

Introducción

El presente Trabajo de Fin de Grado se ha realizado en colaboración con una de las firmas de consultoría y auditoría más importantes a nivel global, en concreto con su departamento de auditoría de sistemas informáticos que trabaja en colaboración con el de auditoría financiera.

El objetivo con el que nace este trabajo es el de especificar y diseñar, en líneas generales, un sistema software de gestión para la documentación de las auditorías de sistemas llevadas a cabo en el departamento en el marco de las auditorías financieras que realiza la firma. El departamento considera que este sistema va a ser altamente necesario en el futuro, esto es debido a que en la actualidad se utiliza un sistema que fue desarrollado inicialmente para las auditorías financieras y que no cubre sus necesidades específicas, si se tiene en cuenta que la demanda en los últimos años de este tipo de auditorías ha ido creciendo y que ha planificado que cada vez van a tener más peso, se hace evidente que tener un sistema propio que permita agilizar su trabajo va a ser clave para poder escalar este tipo de auditorías a más clientes en el futuro.

Para lograr este objetivo se ha dividido el proyecto en dos grandes partes diferenciadas:

La primera parte, correspondiente a la especificación completa del sistema, que es la más crítica y necesaria para el departamento, pues servirá para sustentar parte de la solicitud que se realizará a la dirección para que autorice el desarrollo del sistema.

En primer lugar, se ha realizado un estudio del contexto del sistema, analizando con detalle la situación y el sistema actual, identificando las oportunidades de mejora y analizando los principales *stakeholders* del proyecto. En segundo lugar, se ha procedido a realizar la especificación completa del sistema, definiendo vagamente la arquitectura física que se deberá implementar, definiendo las funcionalidades que necesitarán los auditores y especificando en detalle los requisitos funcionales, en forma de casos de uso, y no funcionales del sistema; y modelizando en UML el modelo conceptual de los datos derivado de esta especificación.

La segunda parte, correspondiente al diseño general del sistema, que se ha realizado a propuesta del propio estudiante ya que en principio no se consideraba necesaria, pero tras comentarlo con la dirección del departamento se dio el visto bueno a incluir un diseño en líneas generales del sistema. Este diseño se articula en tres ejes concretos: arquitectura general del sistema, base de datos e interfaz gráfica.

Para la arquitectura del sistema, se ha definido en líneas generales cómo deberá ser la arquitectura física que de soporte al sistema, qué patrón arquitectónico deberá implementar a nivel lógico y con qué tecnologías se recomienda su implementación.

En cuanto a la base de datos, se ha realizado un análisis para decidir la tecnología que mejor resuelve el problema propuesto, se ha propuesto un esquema en pseudocódigo de una BD que represente los elementos funcionales del sistema y por último se ha analizado el posible uso de índices en las tablas definidas.

Por último y en cuanto a la interfaz gráfica, se ha realizado una propuesta de mapa navegacional del sistema, definiendo las pantallas que deberán ver los usuarios y enlazando cada pantalla con los casos de uso de la especificación a los que deberán tener acceso los usuarios en cada pantalla.

El resultado de este proyecto se ha validado con la dirección del departamento para tener la certeza de que las características del sistema que se estaban definiendo estuvieran alineadas con los objetivos y la visión del propio departamento.

La estructura de la memoria coincide con la estructura mencionada anteriormente, más unos primeros apartados correspondientes a la gestión del proyecto. Se ha separado la fase de especificación y diseño en dos partes de la memoria. La parte I para la especificación y la parte II para el diseño. De esta forma la estructura de la memoria es más clara y delimita mejor el trabajo llevado a cabo para cada fase.

1.- Formulación del problema

1.1- Contexto

El presente trabajo de fin de grado se ha realizado trabajando conjuntamente con una de las principales firmas de servicios profesionales de auditoria y consultoria para empresas. En la división de auditoría de la compañía se encuentra el departamento de SPA (Systems and Processes Assurance), cuyo trabajo consiste en auditar los sistemas de información críticos de las empresas como paso previo a la realización de ciertas partes de la auditoría financiera.

Para realizar estas auditorías, los auditores de este departamento utilizan un sistema software, denominado *Aura*, con el que trabajan a diario y dónde queda registrado todo la documentación y el trabajo relacionado con las auditorías. Desde el departamento se considera que este sistema no cubre correctamente sus necesidades, por lo que se ha planteado una iniciativa interna para sustituir este sistema por uno de nuevo.

Para lograr que esta iniciativa sea aceptada por la dirección, se consideró que disponer de un estudio en el que se analizará el sistema actual y se especificará con detalle que debería ofrecer el sistema para cubrir las necesidades del departamento sería de gran utilidad para fundamentar la solicitud que se realizará a la dirección para que autorice el desarrollo de este sistema. No obstante, debido a la alta carga de trabajo que soportan los miembros del departamento no se encontró ningún integrante del mismo, dentro de los que tienen los conocimientos para realizarlo, con la disponibilidad para realizar este estudio.

Es por esto que se consideró que la realización de un trabajo de fin de grado por parte de un estudiante de la FIB, cuyo objetivo sea el análisis del sistema actual junto con la especificación del nuevo sistema, podría permitir al departamento seguir con su trabajo sin incurrir en un desajuste de su presupuesto y a su vez lograr un informe que permita sustentar la solicitud y que siente las bases del sistema que se necesita desarrollar.

Términos y conceptos de una auditoría de sistemas

La auditoría realizada por el equipo de SPA se centra principalmente en la obtención de evidencias para el testeo de la existencia y la efectividad de los ITGCs (Information Technology General Controls) que las compañías aplican en sus sistemas a auditar. Los ITGCs son una serie de controles estandarizados y definidos por la ISACA (Information Systems Audit and Control Association) y que se aplican sobre los entornos en los que se desarrollan, mantienen y operan estos sistemas, estos controles incluyen las políticas, procedimientos y prácticas establecidas por la dirección de IT de la compañía con el objetivo de asegurar el correcto desarrollo e implementación de las aplicaciones y la integridad de los programas y sus datos. [1]

Una vez se ha documentado la existencia, o falta de ella, y si la implementación de estos controles es eficaz o no, la dirección del equipo de SPA emite un informe interno para los auditores financieros en él que se informa sobre si se puede tener, o no, confianza en la información proporcionada por estos sistemas.

En el caso afirmativo, el trabajo a realizar por los auditores financieros se ve notablemente reducido debido a que la información proporcionada por el sistema no tiene que ser revisada tan exhaustivamente, en caso negativo, los auditores financieros deben realizar muchas más comprobaciones manuales (Contrastar facturas con clientes y proveedores, contactar con bancos para validar importes ingresados…) antes de validar la información del sistema. También se emite un informe para el cliente auditado con un listado de recomendaciones para subsanar controles mal aplicados o la ausencia de ellos.

Problema a resolver

Cómo se puede observar, llevar a cabo una auditoría de sistemas es un proceso en el que la documentación generada es muy importante, debido a que todas las afirmaciones sobre la aplicación de los controles deben estar fundamentadas en las evidencias obtenidas durante el proceso, por lo que una buena gestión de este proceso de documentación es clave para que la auditoría sea lo más ágil posible.

El sistema software actual, desarrollado internamente por la firma, está enfocado principalmente a la auditoría financiera, por lo que muchas funcionalidades no son de ningún uso para la auditoría de sistemas y existen necesidades propias de este tipo de auditorías que no están cubiertas, además, el sistema es antiguo y tiene muchos puntos que se pueden mejorar.

En este trabajo de fin de grado se ha propuesto formalizar la especificación, y parte del diseño, de un sistema software que sirva para la correcta gestión de la documentación propia de una auditoría de sistemas y que pueda sustituir al sistema utilizado actualmente, de forma que se cubran las necesidades específicas del departamento. Para lograr este objetivo, se realizará toda la fase previa necesaria al desarrollo del sistema, que constará de la fase de especificación y análisi de requisitos y de la fase de diseño funcional del sistema. Todas estas fases se realizarán siguiendo las metodologías y buenas prácticas vistas durante el curso de la especialidad de ES del GEI.

Actores implicados

El sistema que se diseñará tendrá como usuarios finales a los propios auditores de sistemas, que utilizarán sus funcionalidades para documentar de una forma más ágil y que les permitirá optimizar mejor el tiempo dedicado a los clientes. Debido a que la gestión del tiempo se verá optimizada, la compañía también se verá beneficiada por el sistema, ya que se espera que el tiempo invertido en la tarea de documentación se vea reducido sustancialmente y como la compañía presupuesta las auditorías en función de las horas necesarias, esta podrá ajustar más los precios o mejorar su margen de beneficio.

Los clientes también se verán beneficiados, actualmente no se realizan auditorias de sistemas en todos los clientes auditados debido a que tiene un coste elevado que solo sale rentable para clientes a partir de un tamaño considerado, ya que con el nuevo sistema se tardará menos en el desarrollo de una auditoría de sistemas y estas se podrán ofrecer a más clientes a un precio más competitivo.

1.2- Justificación

Antes de proceder a especificar el sistema se ha realizado un análisis de posibles alternativas que puedan satisfacer parcial o totalmente las necesidades de los usuarios y del departamento. La valoración de cada opción se realiza a continuación.

Software comercial

En primer lugar se ha contemplado utilizar un software ya desarrollado y que pueda cumplir con las necesidades de la firma, aunque necesite algún desarrollo a medida para adaptarlo, y que solo requiera del pago de este desarrollo y de la propia licencia del software para ser implementado. Se ha realizado una búsqueda exhaustiva y se ha encontrado que existen varios programas destinados a la auditoría informática [2], pero todos ellos destinados a la auditoría interna, control continuado a lo largo del tiempo sobre unos sistemas determinados, y no a la externa, auditoría de varios sistemas de clientes distintos en una duración limitada (entre semanas y meses) que es la desarrollada por la compañía, por lo que se considera que estos sistemas no ofrecen las funcionalidades necesarias y requieren de muchos cambios para ser viables. Además el sistema a desarrollar será crítico para la firma, ya que en él se documentará el trabajo de todas las auditorías, por lo que desde la compañía se prefiere no depender de una empresa externa para el desarrollo, mantenimiento y la resolución de incidencias y se prefiere optar por tener más control sobre el producto que se va a desarrollar, por lo que esta opción queda descartada.

Nueva versión de Aura

En segundo lugar se ha planteado desarrollar desde cero una nueva versión de Aura, el sistema actual , para que dé soporte a las auditorías tanto financieras como informáticas. El principal inconveniente de esta opción es que conlleva tener que identificar todos los requisitos que se necesitan desde la auditoría financiera. Se ha considerado que esta opción queda fuera del alcance de la necesidad que pretende solventar este proyecto, por lo que ha quedado descartada. Aún así el sistema se diseñará de forma que sea fácil reutilizar sus funcionalidades por si en un futuro se quisiera desarrollar un nuevo sistema para la auditoría financiera y se quisiera utilizar esta arquitectura como base.

Módulo de Aura

En línea con la opción anterior, se ha planteado desarrollar un evolutivo de Aura que incorpore un módulo especialmente diseñado para la realización de auditorías de sistemas. Esta opción cubre parte de las necesidades del departamento, pero nos obliga a tener que utilizar la misma tecnología que el sistema actual, de forma que se seguirían teniendo las bases en local en el PC del auditor y se utilizaría la misma interfície. Además, el hecho de que se desconoce como se ha desarrollado el sistema actual supone un gran desafío, ya que no se sabe cómo está diseñado internamente, de qué clases consta, si se han seguido patrones de diseño que permitan que la cambiabilidad del sistema sea buena… Todos estos factores suponen un gran riesgo, y además se estaría desaprovechando una muy buena opción para modernizar la tecnología utilizada, por lo que esta opción queda descartada.

Conclusión

Se observa que las alternativas planteadas han sido todas descartadas por varias razones, ya sea por no cubrir las necesidades del departamento (alternativa *Software comercial*), por tener un alcance demasiado grande (alternativa *Nueva versión Aura*) o por tener una dificultad superior y desaprovechar la oportunidad de modernizar la tecnología utilizada (alternativa *Módulo Aura*). Se puede concluir pues, que no se ha encontrado ninguna opción mejor para resolver el problema planteado

1.3.- Alcance

Desde el departament se nos solicitó que el alcance del proyecto consistiera en la realización de todo el trabajo necesario para la completa especificación de este sistema. Esto incluye el estudio de la situación actual, análisis de puntos fuertes y débiles del sistema actual, identificación de las opciones de mejora que puede aportar el proyecto, análisis y especificación de requisitos; e identificación y descripción de los casos de uso que deberá satisfacer el sistema. Para lograr este objetivo, se consideró que lo más adecuado consistía en combinar el trabajo de auditoría de sistemas, para entender mejor las necesidades del departamento, con la propia elaboración del informe que contenga el estudio realizado.

Adicionalmente, y a petición del propio estudiante, también se ha realizado la fase correspondiente al diseño del sistema, incluyendo el diseño de clases, tablas de la base de datos y una parte de la interfaz.

Todas estas fases se han desarrollado siguiendo las metodologías aprendidas en la FIB en la especialidad de ES. Además, para la correcta especificación del sistema, se ha participado en el desarrollo de varias auditorías de sistemas trabajando como auditor, de forma que se han entendido las necesidades a cubrir de primera mano y ha sido más fácil decidir qué y cómo debería ser el sistema ideal para el departamento.

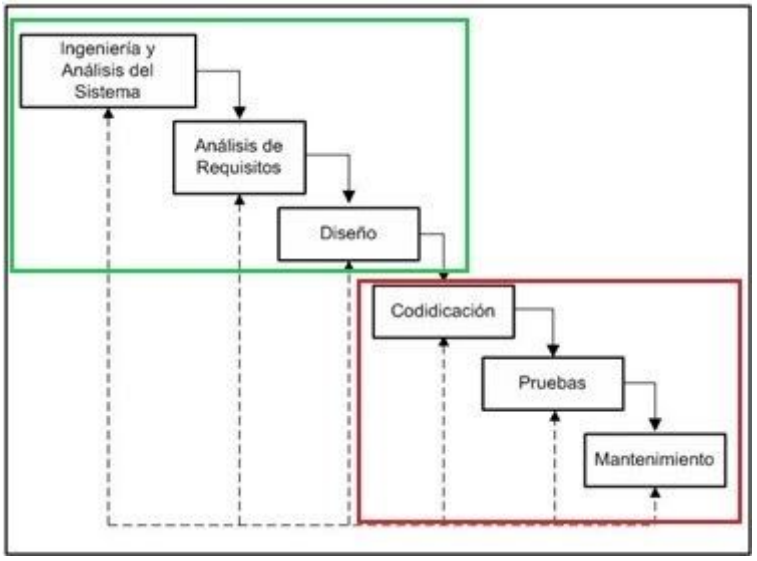


Figura 1. Ciclo de vida del software
(en verde las fases en alcance y en rojo las descartadas)

No ha entrado dentro del alcance la fase correspondiente al desarrollo del sistema, ni tampoco las siguientes fases, ya que debido a la complejidad del mismo y su criticidad la implementación de este sistema queda fuera del alcance de un proyecto de fin de grado. Además, la firma prefiere que la implementación la realice un equipo formado por varios integrantes con más experiencia en el desarrollo de este tipo de software.

1.3.1 - Objetivos del proyecto

Los objetivos y subobjetivos que se han alcanzado con la realización de este proyecto han variado ligeramente respecto a los que se plantearon en la planificación del mismo, ya que en la fase de diseño se han reformulado parte de estos objetivos. Los objetivos finales del proyecto han sido los siguientes:

1. Especificar el sistema

- 1.1. Estudiar situación y sistema actual
  - 1.1.1. Estudiar el contexto en detalle
  - 1.1.2. Modelizar los procesos implicados en el ciclo de vida de una auditoría de sistemas
  - 1.1.3. Identificar puntos fuertes y puntos débiles del sistema actual
  - 1.1.4. Concretar la visión del proyecto, identificar las oportunidades de mejora, analizar alternativas y justificar la decisión tomada.
  - 1.1.5. Identificar los objetivos a cumplir para el sistema
- 1.2. Analizar los requisitos del nuevo sistema
  - 1.2.1. Identificar los stakeholders del nuevo sistema
  - 1.2.2. Identificar y definir los requisitos funcionales del sistema
  - 1.2.3. Identificar y definir los requisitos no funcionales del sistema
  - 1.2.4. Validar con los usuarios los requisitos definidos.
  - 1.2.5. Identificar, definir y analizar los casos de uso del sistema
- 1.3. Especificar el sistema en UML
  - 1.3.1. Identificar las clases del dominio
  - 1.3.2. Definir las relaciones entre clases

2. Diseñar el sistema

- 2.1. Diseñar la arquitectura del sistema
  - 2.1.1. Definir diagrama de diseño en UML
  - 2.1.2. Definir la arquitectura lógica del sistema
  - 2.1.3. Definir la arquitectura física del sistema
  - 2.1.4. Decidir la tecnología que se utilitzarà para implementar el sistema
- 2.2. Diseñar la estructura de la base de datos
  - 2.2.1. Decidir la tecnología a utilizar
  - 2.2.2. Identificar las tablas necesarias
  - 2.2.3. Definir los atributos de cada tabla
  - 2.2.4. Establecer las restricciones necesarias para cada tabla
  - 2.2.5. Definir las relaciones entre tablas
  - 2.2.6. Identificar posibles índices a utilizar para optimizar el rendimiento
- 2.3. Diseñar la interfaz del sistema
  - 2.3.1. Definir el mapa navegacional de las pantallas del sistema

Como se puede observar los objetivos se agrupan en 2 objetivos principales, que corresponden a la especificación del sistema y su diseño.

2.- Planificación

2.1.- Listado de tareas

En la planificación inicial del proyecto, se definieron las tareas que se deberían realizar para alcanzar los objetivos definidos en el alcance del proyecto. Estas tareas se encuentran formalizadas en el punto ***1 - Tabla de tareas*** del **Anexo**, en esta tabla se encuentran estas tareas agrupadas en las distintas fases y subfases de las que consta el proyecto, junto con la estimación inicial en tiempo de cada tarea y las dependencias entre ellas.

Este listado de tareas ha sufrido distintas modificaciones durante el transcurso del proyecto debido a que se han redefinido varios de los objetivos de la fase de diseño y a que no se estimaron correctamente algunas de las tareas de la fase de especificación.

La redefinición de los objetivos de la fase de diseño ha comportado la eliminación de varias tareas, en concreto se se han eliminado las tareas *T25 - Identificar los atributos y las funciones necesarias, T26 - Identificar patrones a utilizar* y *T37 - Diseñar mock-ups de las vistas principales,* al no considerarse necesarias para los nuevos objetivos. La *T25* se ha sustituido por la *T25Alt - Definir a alto nivel la arquitectura lógica del sistema,* con una estimación de 10h, y la *T26* se ha sustituido por la *T26Alt - Decidir la tecnología recomendada para implementar el sistema,* con una estimación de 4h.

En cuanto a las tareas de la fase de especificación cuyo coste fue subestimado, estas corresponden a las subfases *Análisis y especificación de requisitos* y *Especificación del sistema en un diagrama de clases,* se concretan en el punto siguiente que trata sobre la planificación temporal debido a que el efecto que ha tenido sobre la planificación su subestimación.

Respecto a las dependencias entre tareas, durante la ejecución del proyecto se ha considerado que las tareas de las subfases *Análisis y especificación de requisitos* y *Especificación del sistema en un diagrama de clases* no tenían realmente una dependencia temporal entre ellas, por lo que se han realizado de forma paralela.

El coste de las otras tareas se ha aproximado bastante, o incluso han costado menos, a lo estimado y sus dependencias se han mantenido tal y cómo se planificaron, por lo que se considera que la planificación para el resto del proyecto ha sido buena. Cabe destacar que la tarea *T38 - Definir mapa navegacional,* se ha realizado en la mitad de tiempo que se había planificado inicialmente, por lo que su coste final se ha reducido a la mitad, de 20h a 10h.

2.2.- Planificación temporal

La planificación actual del proyecto ha sufrido ligeras desviaciones respecto a lo planeado, pero aún así el proyecto ha podido terminar en una fecha razonable, aunque para esto se han tenido que redefinir y recortar algunos de los objetivos planteados para la fase de diseño, de forma que la carga de trabajo en estas tareas se ha visto reducida para poder llegar a tiempo y cumplir con la fecha límite del proyecto.

Para analizar las desviaciones que han tenido lugar, se comparan las distintas fechas de vencimiento que se habían planificado para las partes que conforman cada una de las dos fases del proyecto con la fecha de terminación real de estas fases (para las que ya se encuentren terminadas) y con la nueva fecha de terminación planificada a dia de hoy.

La planificación inicial se puede encontrar en el punto ***2 - Diagrama de Gantt*** del **Anexo**, la comparación se realiza en la siguiente tabla:

Fase	**Subfase**	**Fecha terminación planificación inicial**	**Fecha terminación real / planificada actualmente**
Especificación	Estudio de la situación y sistema actual	09/03/2020	**Finalizada el 06/03/2020**
	Análisis y especificación de requisitos	30/03/2020	**Finalizada el 24/04/2020**
	Especificación del sistema en un diagrama de clases	10/04/2020	**Finalizada el 24/04/2020**
Diseño	Diseño de la arquitectura del sistema	27/04/2020	**Finalizada el 10/05/2020**
	Diseño de la estructura de la BD	12/05/2020	**Finalizada el 28/05/2020**
	Diseño de la interfaz del sistema	1/06/2020	**Finalizada el 12/06/2020**
	Redacción de la memoria	08/06/2020	**Finalizada el 16/06/2020**

Figura 2. Tabla de planificación de las partes del proyecto donde se puede comparar su fecha de terminación planificada inicialmente respecto a la fecha de terminación real.

Como se puede observar, el proyecto se ha retrasado respecto a lo que se planificó inicialmente, pero aún así la fecha de finalización del proyecto (18/06/2020) ha entrado dentro del término para la presentación de la memoria, por lo que se podrá entregar y defender sin mayores consecuencias.

En la tabla también se evidencia que el retraso del proyecto se ha registrado principalmente en dos apartados concretos de la fase de especificación, correspondientes al *Análisis y especificación de requisitos* y a la *Especificación del sistema en un diagrama de clases.* Ya que se encontraban planificados para el 30/03/2020 y el 10/04/2020 respectivamente pero han sido finalizados los dos en el 24/04/2020.

Este retraso en estos dos apartados se ha debido a que no se estimó correctamente la complejidad que entrañaba la especificación de un sistema complejo y con muchos casos de uso distintos, por lo que se planificó una fecha de finalización para nada correspondiente con la realidad.

Además, se planificaron ambos apartados como unidades independientes, pero a la práctica las tareas de estos apartados se han ido realizando de forma simultánea. Esto ha sido debido a que a medida que se iban especificando los requisitos del sistema también se iba especificando y modelando el diagrama de clases para que se ajustase a los requisitos que debía cumplir, esta forma de operar ha permitido realizar la especificación de forma iterativa, asegurando así que todas las funcionalidades han sido cubiertas y que la especificación del diagrama de clases permite cubrir los requisitos indicados.

Las tareas de esta fase cuyo coste se ha desmarcado de forma más pronunciada han sido las siguientes:

Subfase - Análisis y especificación de requisitos

- **T11.1 - Identificar los requisitos funcionales (12h -> 20h):** En esta tarea se han analizado y definido todas las funcionalidades que debe cubrir el sistema. En un principio fue estimada en 12h, pero a la práctica se han encontrado con un gran surtido de funcionalidades que se debían analizar y definir con detalle, de forma que esta tarea ha acabado costando unas 20h aproximadamente.

- **T11.2 - Identificar los requisitos funcionales (2h -> 10h):** Definir los requisitos no funcionales se estimó en 2h, una estimación que fue demasiado optimista, ya que a la práctica el sólo hecho de buscar documentación fiable y analizar con detalle qué requerimientos aplican para este sistema ha implicado mucho más de estas 2h. Al final, se estima que esta tarea ha costado unas 10h aproximadamente.

- **T13 - Identificar, analizar y definir los casos de uso del sistema (20h -> 60h):** Sin lugar a duda, esta ha sido la tarea que más tiempo ha llevado y que más se subestimó en la planificación. El hecho de que hayan salido más de 100 casos de uso distintos del sistema como parte de su especificación, que cada caso de uso se ha tenido que analizar y valorar para luego definirlo, ha hecho que las 20h planificadas se hayan quedado extremadamente cortas con el coste real de la misma. Se estima que esta tarea ha costado unas 60h aproximadamente.

Subfase - Especificación del sistema en un diagrama de clases

- **T19 - Plasmar las clases y las relaciones en un diagrama UML (2h -> 10h):** La estimación inicial se realizó teniendo en mente de que la definición del UML se haría una sola vez y sería única, pero durante el transcurso del proyecto se ha encontrado que se han tenido que ir definiendo nuevas relaciones y redefiniendo antiguas a medida que se avanzaba con la especificación de la misma, por lo que se ha tenido que ir rehaciendo este diagrama de forma que las 2h planificadas inicialmente se han convertido en unas 10h durante el transcurso de esta subfase.

En total, se ha producido un sobrecoste de 64h respecto a lo planificado. En cuanto a la fase de diseño, todos los apartados se han realizado relativamente acorde a su planificación y no se ha producido ningún retraso en esta fase, pero se tiene que tener en cuenta que se ha habido una reducción del alcance en esta parte, por lo que el hecho de que no se hayan producido retrasos es en parte debido a que se ha reducido la carga de trabajo de esta fase.

2.3.- Presupuesto

El importe presupuestado inicialmente para el proyecto se realizó teniendo en cuenta las horas planificadas en la tabla de tareas anterior y el precio por hora que costaría contratar a un profesional para que realizara estas tareas, esta estimación realizada inicialmente se situaba entorno a los **7125 euros** con el coste de los recursos humanos más las partidas de contingencia como principales elementos.

Teniendo en cuenta los sobrecostes presentes en ciertas tareas de la fase de especificación y la redefinición/eliminación de varias tareas de la fase de diseño, se ha recalculado cual hubiera sido el coste real de este proyecto. El coste final del proyecto equivale al coste del personal, ya que los costes de recursos materiales y recursos softwares son nulos debido a que el portátil usado era reutilizado y a que el software utilizado para el proyecto no ha derivado en ningún coste en forma de licencia.

A continuación, se adjunta la tabla en la que se calcula el coste de los recursos humanos de este proyecto, para calcular los precios de las horas se han usado los mismos importes que en la estimación del presupuesto:

Subfase	Tiempo (h)	Rol	Precio por hora	Coste presupuestado (en euros)
Estudio de la situación y sistema actual	54	Project Manager	24	1296
Análisis y especificación de requisitos	96			2304
Especificación del sistema en un diagrama de clases	18			432
Diseño de la arquitectura del sistema	22	Arquitecto de Software	28,5	627
Diseño de la estructura de la BD	18			513
Diseño de la interfaz del sistema	16	UI/UX Designer	19	304
Total horas	224	-	-	5476
Coste SS	253	-	Un 30% del coste total de las horas (Aprox)	1642,8
Total RH				7118,8

Figura 3. Tabla con el coste final de los recursos humanos del proyecto

Se puede observar como el coste final del proyecto se habría situado en **7118 Euros**, un valor que se encuentra dentro de lo presupuestado inicialmente. Aunque se tiene que tener en consideración que se ha recortado parte del alcance de la fase de diseño para compensar el sobrecoste incurrido en la fase de especificación, si se tiene en cuenta que no se tenía experiencia previa en ningún trabajo real de este tipo, se puede considerar todo un éxito que el presupuesto se haya ajustado al importe presupuestado aunque se hayan tenido que redefinir los objetivos de la fase de diseño.

3.- Metodología y rigor

Para la realización del proyecto, cuyo resultado final ha consistido en un informe con la especificación y el diseño general de un sistema software aplicado a la realización de auditorías informáticas, se han aplicado varias de las metodologías y buenas prácticas aprendidas en la facultad durante el curso de la especialidad de ES del GEI, para así tener más certeza de que la solución obtenida es una solución de calidad.

Solución propuesta

La fase de especificación se ha llevado a cabo siguiendo la misma estructura que se recomienda en el proyecto de la asignatura de *Enginyeria de Requisits,* que consiste en la especificación de un sistema fictício. Se ha decidido seguir la misma estructura de ese proyecto en este, ya que se consideró que seguir un modelo que ha sido revisado por los docentes de la asignatura de ER era la mejor forma de cubrir todos los puntos necesarios para una buena especificación.

Al seguir este modelo se ha realizado primero un análisi del contexto y del sistema actual que usa la compañía, se ha planteado la visión del proyecto y las oportunidades de mejora; y se han analizado los *stakeholders* interesados en el proyecto. Luego se ha realizado la propia especificación, definiendo en primer lugar la visión general del sistema a desarrollar, analizando qué arquitectura física se quiere para el sistema y definiendo a alto nivel las funcionalidades de las que deberá disponer para cubrir las necesidades de los auditores; y definiendo en segundo lugar los propios requisitos del sistema, comprendidos por los requisitos funcionales definidos mediante casos de uso y los requisitos no funcionales.

Para la fase de diseño inicialmente se quería realizar un diseño muy detallado en tres ámbitos distintos del sistema: arquitectura lógica, base de datos e interfaz gráfica. Se había planteado definir todas las clases, atributos, métodos a implementar y patrones de diseño a utilizar en su arquitectura, juntamente con el diseño de toda la base de datos y del diseño de mockups de todas sus pantallas junto con un mapa navegacional para la interfaz.

Durante el transcurso del proyecto, al ver la alta complejidad del sistema derivada de la gran cantidad de casos de uso del mismo y el sobrecoste en horas invertido en la especificación, se tuvo que replantear el alcance de esta fase y realizar un diseño en líneas más generales y que su vez fuera más flexible para que el equipo encargado de su implementación pueda partir de este.

El resultado final de la fase de diseño consta de las siguientes partes:

- **Diseño arquitectónico:** Se ha realizado un diseño general de la arquitectura que debería implementar el sistema para poder desplegarse en la nube y ser accedido por los usuarios vía una aplicación web, tal y como se planteó con el departamento. Este parte ha consistido principalmente en la definición de su arquitectura física y en la implementación del patrón MVC, ambas partes realizadas aplicando los conocimientos aprendidos en la asignatura de *Aplicacions i Serveis Web.* También se han analizado las tecnologías a utilizar y se han recomendado aquellas que se han considerado más adecuadas.

- **Diseño BD:** Se ha partido del diagrama de clases en UML generado en la especificación y se han convertido a un diagrama de clases UML de diseño, modificando las asociaciones y clases que no podían ser implementadas en código y también añadiendo la navegabilidad de la que deberán disponer estas clases, ambos diagramas y su conversión se ha realizado utilizando las técnicas y los conocimientos aprendidos en la asignatura de *Arquitectura del Software*. Luego se ha analizado la tecnología más adecuada para resolver el problema planteado aplicando los conocimientos aprendidos en la asignatura de *Bases de dades.* Por último, con el diagrama UML de diseño obtenido anteriormente se han representado sus clases, atributos y relaciones en forma de tablas, campos, claves y restricciones SQL, generando el esquema de una base de datos compatible con el modelo de clases obtenido, para luego analizar los posibles índices a aplicar en estas tablas para optimizar su rendimiento. Esta última parte se ha realizado aplicando los conocimientos de la asignatura de *Disseny de Bases de Dades.*

- **Diseño interfaz:** En este apartado se ha realizado una simple aproximación a las pantallas que necesitará el sistema y que funcionalidades y casos de uso serán accesibles desde cada una. Para lograrlo, se ha diseñado un mapa de navegabilidad de la aplicación web del sistema y luego se han definido los detalles de cada una de las pantallas representadas en este diagrama.

Validación de la solución

La validación del resultado final, tanto de la especificación como del diseño, es altamente subjetiva al tratarse de una propuesta de sistema que se deberá de implementar a posteriori, de forma que no existe ninguna métrica ni datos que puedan dar información objetiva del resultado de la solución. Para evitar que esta subjetividad pudiera llevar a una solución incorrecta y que esta no cubriera las necesidades de los auditores debido a una mala interpretación del problema, se han validado todas las fases junto con la Sra. Andrea Bianchimano, Ingeniera Informática y gerente del departamento con más de 10 años de experiencia en el sector, para de esta forma asegurar que el sistema especificado se ajustase correctamente a las necesidades del departamento y que la propuesta de diseño general estuviese siempre en línea con el tipo de sistema demandado por el departamento.

4.- Informe de sostenibilidad

En todo proyecto de ingeniería es importante realizar una evaluación de sostenibilidad en los tres aspectos principales: dimensión económica, dimensión social, y dimensión ambiental. No se puede desarrollar un proyecto sin tener en cuenta las implicaciones que éste tendrá sobre la sociedad y el entorno en el cual se desarrolla.

4.1.- Informe de autoevaluación

Después de realizar la encuesta propuesta, creo que en la FIB se aprende lo justo sobre sostenibilidad económica, ambiental y social. Me he dado cuenta que se nos ha concienciado sobre el concepto de la sostenibilidad en sus distintos ámbitos, más en los ámbitos económicos y ambientales que no tanto sociales, pero estos conceptos se quedan a un nivel muy teórico en mi opinión. Además no opino que se nos hayan enseñado a utilizar indicadores para medir la sostenibilidad de un proyecto que no sean económicos, por lo que el criterio que acostumbramos a seguir para valorar la sostenibilidad social y ambiental es la del sentido común.

Personalmente creo que en el mundo en el que vivimos, en el momento que nos planteamos el desarrollo de un proyecto el ámbito de la sostenibilidad que siempre se prioriza es económico y se dejan más de banda las otras dos dimensiones. Esto es debido a que se quiere que el desarrollo sea viable económicamente y la gran mayoría de desarrollos tienen como objetivo generar un beneficio económico con su implementación, por lo que esa sostenibilidad es clave. La sostenibilidad social en estos casos también se valora, pero porque normalmente una proyecto que mejore la calidad de vida de las personas tiene un potencial económico intrínseco. Por último creo que la dimensión medioambiental de la sostenibilidad que muy ignorada en los proyectos, al menos los relacionados con software, debido a que al no generar ningún producto físico tendemos a olvidar que nuestro desarrollo pueda tener impacto en este aspecto.

En conclusión, creo que aún se puede mejorar mucho respeto a la sostenibilidad en el mundo de la informática, empezando a priorizar más la sostenibilidad medioambiental y social y no tanto la económica. Por lo que creo que es importante que se traten todos estos ámbitos desde la facultad.

4.2.- Dimensión ambiental

Proyecto puesto en producción

Los únicos recursos que han sido necesarios para la realización de este proyecto han consistido en un PC para la documentación de las tareas realizadas y en varias plataformas de Software as a Service, también llamado SaaS, que se encuentran alojadas en distintos los servidores del Cloud. Al no haberse utilizado otros recursos, el único impacto ambiental que ha tenido la realización de este proyecto ha consistido en el consumo energético del PC utilizado y el consumo parcial (ya que su potencia de cálculo y por ende su consumo se reparten entre todos los usuarios del servicio) de los servidores que ejecutan el Software en el Cloud.

Se ha considerado que estos consumos son prácticamente irrisorios, por lo que el proyecto tiene muy poco impacto ambiental, además este pequeño impacto se ha intentado mitigar utilizando un PC reutilizado para las tareas de documentación. Al ser el impacto tan bajo, no se ha encontrado ninguna alternativa para realizar el proyecto con una huella ambiental menor.

Vida útil

La vida útil de este proyecto no tiene sentido si no se realiza un posterior desarrollo que implemente el sistema especificado, por lo que en este apartado se asume que en el futuro se desarrollará el sistema y se analiza la vida útil de ese sistema.

Los principales recursos que se van a consumir durante la vida útil del sistema serán recursos energéticos, consumidos en forma de electricidad, tanto por el consumo derivado de la propia infraestructura que hospedará el sistema como por el derivado de parte del consumo de los PCs de los auditores al conectarse a este sistema. El impacto ambiental de estos dependerá del consumo de sus componentes, pero estos son desconocidos actualmente, por lo que ahora mismo no se puede calcular la huella ambiental que tendrá la vida útil del sistema.

Lo que sí se puede afirmar con gran certeza es que la huella ambiental del sistema será menor a la huella ambiental del sistema que se usa en la actualidad, ya que al ejecutarse íntegramente en el cloud el consumo de la aplicación en los portátiles de los auditores se reducirá en gran parte, debido a que no se almacenarán ni se procesarán los datos en local y no se deberán ejecutar operaciones costosas en estos portátiles, sino que solo se realizarán solicitudes simples al servidor en el Cloud, que es el que procesará toda la información. Este servidor estará diseñado para responder a estas peticiones y utilizará los recursos de forma eficiente, por lo que se espera que en global el consumo energético asociado al uso de este sistema sea menor al del sistema actual.

Riesgos

No se observa ningún riesgo relacionado con la huella ambiental del proyecto, como máximo se podría argumentar que si su implementación acabase utilizando muchos recursos esta huella podría aumentar, pero es una posibilidad difícil de cuantificar tanto en su impacto ambiental como en las probabilidades de que ocurra.

4.3.- Dimensión económica

Proyecto puesto en producción

El coste final del proyecto se ha determinado detalladamente en el punto *2.3.- Presupuesto,* dónde se observa que ha sido de **7118 Euros**. Este coste final se deriva únicamente del coste de los recursos humanos que han intervenido en el proyecto, ya que no se han incurrido en costes materiales. Al estar todos los costes centralizados en el personal, no se ha encontrado ninguna forma de ahorrar en la factura, ya que reducir este coste implicaría reducir el número de horas dedicadas al proyecto, por lo que se estaría sacrificando parte de la calidad por un menor coste, y esta opción no se ha considerado en ningún momento.

También se puede apreciar que el coste final del proyecto se ha ajustado a lo presupuestado inicialmente, esto ha sido posible ya que inicialmente se presupuesto un margen extra como contingencia para imprevistos, el cual ha sido vital para amortizar el extra de horas que se ha tenido que invertir en la especificación del sistema y que no se tuvieron en cuenta en el inicio.

Por último, remarcar que este coste corresponde únicamente a este proyecto consistente en la especificación y diseño del sistema, por lo que para disponer del sistema operativo se debería presupuestar y añadir el coste derivado de su posterior implementación.

Vida útil

La vida útil de este proyecto no tiene sentido si no se realiza un posterior desarrollo que implemente el sistema especificado, por lo que en este apartado se asume que en el futuro se desarrollará el sistema y se analiza la vida útil de ese sistema.

Al ser un sistema software que se utilizará para dar soporte a una actividad económica, las auditorías de sistemas, a medida que vayan cambiando las necesidades específicas de esta actividad y del departamento que la realiza el sistema deberá adaptarse a ellas y ser actualizado para seguir proporcionando valor al departamento. Los costes de estas actualizaciones dependerán de la periodicidad y de la magnitud de los cambios que se introduzcan en ellas, por lo que es difícil de prever cual podrá ser el coste asociado a estos cambios. También se debe de considerar el coste del mantenimiento, tanto de la infraestructura hardware utilizada por el sistema como del soporte software que necesiten los auditores si se dan errores o problemas con este. Estos costes dependerán de las necesidades del sistema una vez implementado, por lo que se ha considerado arriesgado realizar una estimación con tanta poca información de antemano.

Lo que sí que se ha considerado es cómo se podría reducir este coste a lo largo de la vida útil del proyecto, y ambas propuestas radican en el desarrollo de este. En primer lugar, una buena implementación a nivel de la arquitectura del código será crucial para que el sistema tenga una buena cambiabilidad y pueda ser actualizado con mayor facilidad, reduciendo así el coste. También, y en este punto entra el proyecto realizado, implementar el sistema utilizando una tecnología actual y una arquitectura física pensada para el futuro, aprovechando el potencial del *cloud*, permitirá que el sistema tenga más vida útil y que no sea necesario cambiar su arquitectura en mucho tiempo.

Riesgos

El principal riesgo con el que cuenta este proyecto es la denegación de su implementación por parte de la dirección de la firma debido al coste ligado a su implementación. Esta posibilidad no es para nada descabellada en la situación actual, en medio de una pandemia global en la que las inversiones no prioritarias han quedado relegadas a la espera de ver cómo evoluciona la situación económica y el impacto que tendrá en los balances de la compañía. La única forma de mitigar este riesgo que se ha encontrado sería la de posponer su implementación para un futuro próximo, ya que el efecto de la pandemia es irreversible, por lo que no se puede hacer nada en este aspecto.

4.4.- Dimensión social

Proyecto puesto en producción

A nivel personal, este proyecto a resultado en una buena experiencia laboral en la que he podido aplicar los conceptos relacionados con la especificación y el diseño de sistemas software en una situación del mundo real fuera de la burbuja de la universidad. He podido ver lo Espero aprender cómo aplicar los conceptos aprendidos en las asignaturas de la especialidad de ES en el proyecto y de esta forma realizar una especificación y un diseño del sistema de calidad.

Vida útil

La vida útil de este proyecto no tiene sentido si no se realiza un posterior desarrollo que implemente el sistema especificado, por lo que en este apartado se asume que en el futuro se desarrollará el sistema y se analiza la vida útil de ese sistema. En este proyecto se ha tratado de ofrecer una especificación y un diseño en líneas generales del sistema de calidad para facilitar su futura implementación.

La solución propuesta en este sistema tendrá un impacto social relativamente bajo, debido a que es un sistema principalmente de gestión y pensado para optimizar la realización del trabajo de documentación de auditorías informáticas, por lo que no impactará demasiado en en la calidad de vida de sus usuarios, ya que solamente mejorará y facilitará su forma de trabajar.

Riesgo

No se considera que el sistema pueda terminar siendo perjudicial para ninguna parte de la población, ya que básicamente es un sistema que se usará de forma interna por la compañía y su uso no afectará a nadie de fuera de esta. Además su único propósito es beneficiar al departamento, facilitando la realización de auditorías informáticas, por lo que si este beneficio desapareciera también lo haría el motivo de existencia del sistema.

Tampoco se considera que los usuarios puedan desarrollar algún tipo de dependencia hacia este sistema que los ponga en una posición de debilidad futura, ya que este sistema es solo una herramienta que la compañía pone a su disposición y en caso de que desapareciera solamente debería ser sustituida por otra para que los auditores pudieran seguir realizando su trabajo sin complicaciones.

4.5.- Conclusiones

Una vez he analizado y reflexionado sobre la matriz de sostenibilidad del proyecto, considero que entiendo los índices de sostenibilidad y estoy capacitado para medir cómo puntúa el proyecto en cada uno de ellos.

En líneas generales, creo que el proyecto tiene un buen nivel de sostenibilidad para cada uno de las tres dimensiones que se han tenido en cuenta para su evaluación.

En la dimensión ambiental, considero que el proyecto tiene una huella ecológica relativamente baja al tratarse de un sistema software que solamente consume recursos energéticos en forma de electricidad. Si analizamos los recursos necesarios a nivel de hardware, podemos observar que la compañía aplica una política de reutilización de los dispositivos de los auditores para trabajar (de forma que se reduce la necesidad de disponer de nuevos equipos) y que el sistema se hospedará en la infraestructura de un proveedor Cloud, la cual será también aprovechada por múltiples clientes y proyectos distintos (por lo que no será necesario desplegar nuevo hardware específicamente para este sistema). Si tenemos en cuenta estos puntos, y que actualmente se utiliza un sistema mucho menos óptimo en el uso de recursos, se puede considerar que la huella ecológica del sistema es buena y mejorará la huella del sistema actual.

A nivel económico, he constatado lo difícil que puede ser hacer un presupuesto ajustado que sea acorde a la realidad, ya que el coste final del proyecto ha entrado muy justo con el planificado, en gran parte gracias a las partidas extras de contingencia que se introdujeron en esta planificación. También he reflexionado sobre el riesgo que tiene el desarrollo del sistema actualmente, ya que su implementación implica un coste que puede que la dirección de la firma no quiera asumir en este periodo de incertidumbre. Aún así, considero que el sistema servirá para mejorar la productividad del departamento y que este tipo de inversión puede ser muy beneficiosa en el largo plazo. Por último, si la implementación del sistema se realiza aplicando una buena arquitectura y se utilizan tecnologías que se recomiendan en el diseño, creo que el coste de mantenimiento y actualización del sistema no será elevado, por lo que su viabilidad económica a largo plazo aún se ve más reforzada.

A nivel social, no creo que el sistema tenga un gran impacto en las vidas de sus usuarios, pero lo que es seguro es que no tendrá ningún impacto negativo en estos, ya que su principal motivo para existir es facilitar el trabajo y cubrir mejor las necesidades de los integrantes de este departamento. Es por esto que se considera que el impacto social será positivo, al facilitar el trabajo de sus usuarios, pero muy limitado, ya que es un producto interno de la compañía que sólo llegará a un usuario muy específico y que sólo le beneficiará en su ámbito laboral.

Al tener en cuenta las tres dimensiones, creo que el sistema no tiene puntos negativos en ninguna de ellas, tampoco es excepcionalmente sostenible al no destacar en ninguna, pero el líneas generales se puede afirmar que este proyecto, junto con el proyecto encargado del diseño detallado y la implementación del sistema, serán sostenibles a nivel ambiental, económico y social.

Parte I

-

Fase de especificación

Estudio del contexto

Para entender qué sistema se tiene que desarrollar, primero se debe entender el contexto de la compañía: cuál es su negocio y a quién se dirige, como trabajan en la actualidad, qué necesidades tienen qué les va a cubrir el nuevo sistema, para qué van a utilizarlo, quienes serán los usuarios que le van a dar uso… Entender y dar respuesta a todas estas cuestiones es clave para especificar correctamente el sistema, por lo que en este apartado se van a tratar en profundidad.

5.- Situación actual

En primer lugar, se va a estudiar la situación actual de la compañía. Se analizará y entenderá en qué consiste su negocio, cuál es el problema que se plantea y cómo se está operando en la actualidad.

5.1- Análisis y entendimiento de la situación actual

Conceptos clave

Antes de empezar el estudio del contexto, es importante definir todo un seguido de conceptos específicos del dominio y que son necesarios para entenderlo correctamente. Los conceptos identificados como clave son los siguientes:

- **Base:** Concepto del sistema que se corresponde directamente a una auditoría concreta para un cliente, en la base se almacena toda la información relacionada con esta auditoría y se documentan todos los aspectos relevantes de esta. A partir de la información almacenada en la base se fundamentan los informes que se emitirán al final de la auditoría, es decir, todas las afirmaciones realizadas en estos informes deben de estar respaldadas por la información que contiene la base.

- **ITGC:** También denominados Controles Generales Informáticos. Son controles estandarizados definidos por la ISACA que se aplican sobre los entornos en los que se desarrollan, mantienen y operan los sistemas que se auditan. Estos controles incluyen las políticas, procedimientos y prácticas establecidas por la dirección de IT de la compañía con el objetivo de asegurar el correcto desarrollo e implementación de las aplicaciones y la integridad de los programas y sus datos.

- **Área de aplicación:** Los ITGCs definidos anteriormente se aplican en distintos ámbitos de los sistemas, desde la compañía se han identificado 5 ámbitos principales para los que se testean los controles que son los siguientes: Entendimiento del entorno IT, Seguridad, Gestión de cambios, Operaciones informáticas y Desarrollo de programas.

- **Evidencia:** Se denomina evidencia al documento que proporciona una información veraz e inequívoca del estado, durante el período auditado, de la configuración de los sistemas de información o de las políticas que les apliquen. Esta información puede ser obtenida en distintos formatos y de diversas fuentes, algunos ejemplos de evidencia son los siguientes: capturas de pantalla con la información mostrada por el sistema, ficheros de configuración, el contenido de las tablas de una base de datos extraído en formato Excel adjuntando la consulta realizada en SQL...

- **Requerimientos de información:** Describen exactamente qué condiciones deben cumplir las evidencias que se solicitarán y en qué formato se deben entregar, se agrupan en un listado y cada punto representa un aspecto a auditar (por ejemplo, la política de configuración de contraseñas de un cierto sistema)

- **Servicio de Directorio (SD):** Es una aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red de ordenadores y sobre los recursos de red y que permite a los administradores gestionar el acceso de los usuarios a los recursos sobre dicha red. [3] Esto permite a los usuarios acceder a su PC utilizando el usuario y contraseña registrados en el SD de la compañía y de esta forma acceder a los servicios y aplicaciones para los que tengan autorización sin necesidad de introducir sus credenciales de nuevo.

Sistema actual

Actualmente, para la documentación se utiliza un sistema software denominado “Aura”, desarrollado internamente por el grupo del que forma parte la compañía, que está enfocado principalmente a la auditoría financiera, por lo que muchas funcionalidades no son de ningún uso para la auditoría de sistemas y existen necesidades propias de este tipo de auditorías que no están cubiertas. El sistema está formado por un repositorio central de bases que se encuentra en la nube y una aplicación de escritorio instalada en los PCs de los auditores que replica estas bases y las copia íntegramente en el disco duro de cada PC.

Cada base del sistema corresponde a una auditoría y dentro de cada base se realizan a la par la auditoría financiera y la auditoría de sistemas en caso de que esta sea necesaria. En estas bases existe el concepto “Control”, que para SPA corresponde con el concepto de ITGC, y para cada control se crea uno o varios EGAs, en el caso de que el control sea el mismo para varios sistemas se crea un EGA asociado al mismo control para cada sistema.

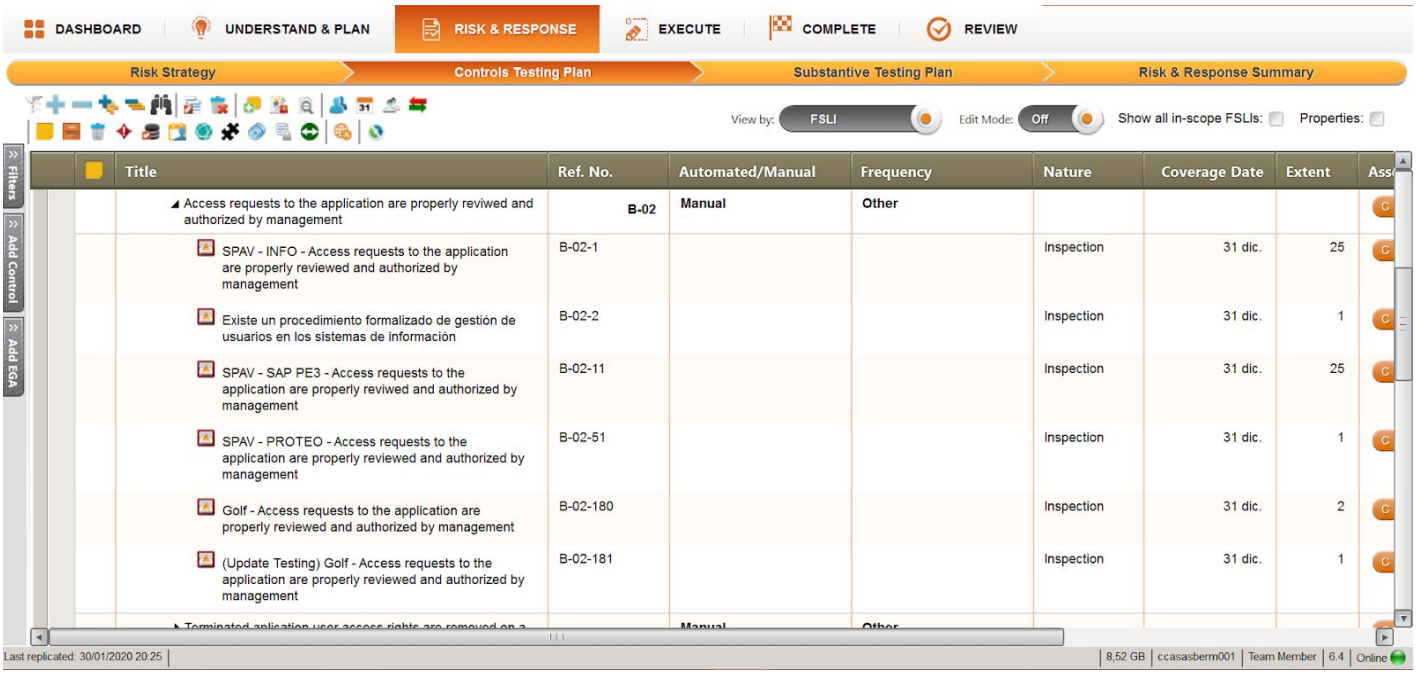
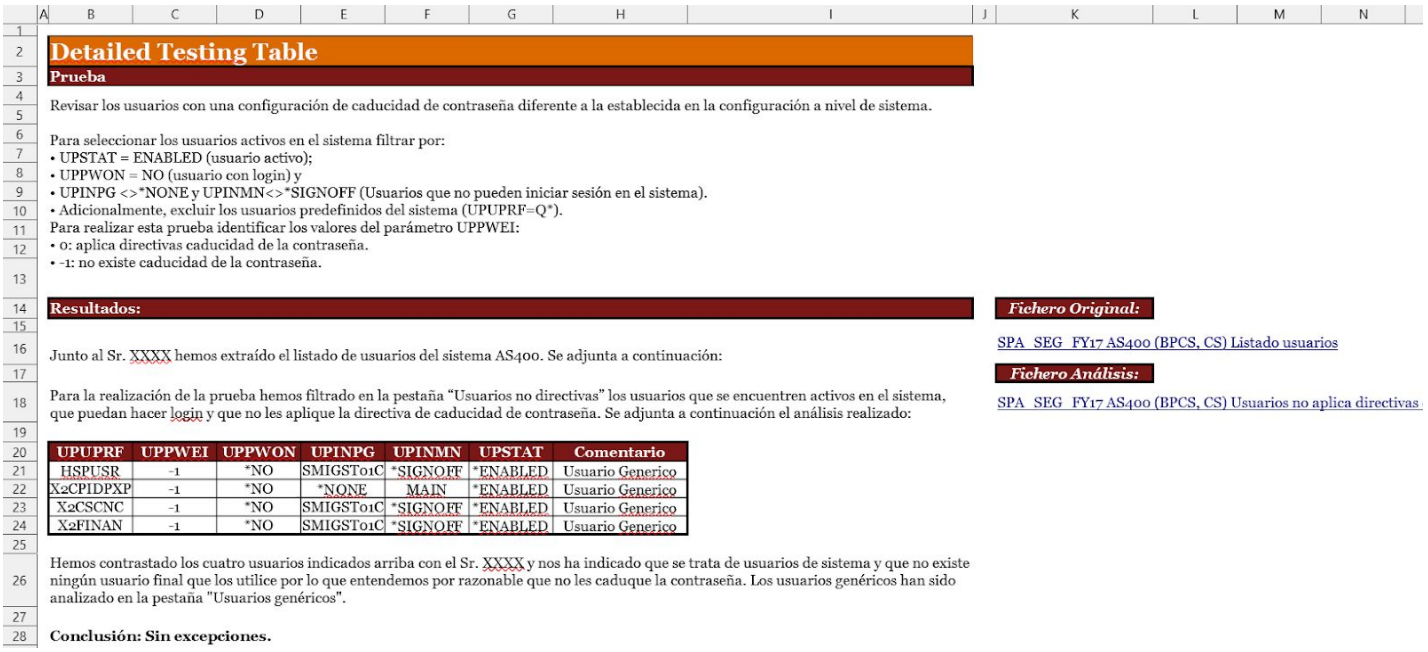


Figura 4. Ejemplo de un control del área de Seguridad con varios EGAs para los distintos sistemas

Los EGAs corresponden siempre a ficheros en formato *xlsx,* o tipo "Excel", y en ellos se documentan el testeo de los ITGCs establecidos para los sistemas de los clientes. Cada EGA consta de varias pestañas, en la que se testea un ámbito distinto del control. Por ejemplo, un EGA que documenta el ITGC "Passwords and security configurations are set in an effective manner" consta de varias pestañas, en una se analizan las directivas de contraseñas, en otra se analizan los usuarios y se comprueba que estas políticas les apliquen, y en la última se analizan los usuarios genéricos presentes en el sistema. A continuación se muestra un ejemplo de una de estas pestañas:



Detailed Testing Table

Prueba

Revisar los usuarios con una configuración de caducidad de contraseña diferente a la establecida en la configuración a nivel de sistema.

Para seleccionar los usuarios activos en el sistema filtrar por:
- UPSTAT = ENABLED (usuario activo);
- UPPWON = NO (usuario con login) y
- UPINPG <>*NONE y UPINMN<>*SIGNOFF (Usuarios que no pueden iniciar sesión en el sistema).
- Adicionalmente, excluir los usuarios predefinidos del sistema (UPUPRF=Q*).

Para realizar esta prueba identificar los valores del parámetro UPPWEI:
- 0: aplica directivas caducidad de la contraseña.
- -1: no existe caducidad de la contraseña.

Resultados:

Fichero Original:

SPA_SEG_FY17 AS400 (BPCS, CS) Listado usuarios

Fichero Análisis:

SPA_SEG_FY17 AS400 (BPCS, CS) Usuarios no aplica directivas

Junto al Sr. XXXX hemos extraído el listado de usuarios del sistema AS400. Se adjunta a continuación:

Para la realización de la prueba hemos filtrado en la pestaña "Usuarios no directivas" los usuarios que se encuentren activos en el sistema, que puedan hacer login y que no les aplique la directiva de caducidad de contraseña. Se adjunta a continuación el análisis realizado:

UPUPRF	UPPWEI	UPPWON	UPINPG	UPINMN	UPSTAT	Comentario
HSPUSR	-1	*NO	SMIGST01C	*SIGNOFF	*ENABLED	Usuario Generico
X2CPIDPXP	-1	*NO	*NONE	MAIN	*ENABLED	Usuario Generico
X2CSCNC	-1	*NO	SMIGST01C	*SIGNOFF	*ENABLED	Usuario Generico
X2FINAN	-1	*NO	SMIGST01C	*SIGNOFF	*ENABLED	Usuario Generico

Hemos contrastado los cuatro usuarios indicados arriba con el Sr. XXXX y nos ha indicado que se trata de usuarios de sistema y que no existe ningún usuario final que los utilice por lo que entendemos por razonable que no les caduque la contraseña. Los usuarios genéricos han sido analizado en la pestaña "Usuarios genéricos".

Conclusión: Sin excepciones.

Figura 5. Ejemplo de una pestaña de un EGA para el ITGC del área de Seguridad "Passwords and security configurations are set in an effective manner"

Cuando el auditor realiza modificaciones o añade información a la base, este se encuentra trabajando sobre una copia en local, por lo que estas modificaciones no son visibles para el resto del equipo hasta que no replique contra el servidor central. En cuanto selecciona esta opción, estas modificaciones son sincronizadas con el repositorio de forma que quedan registradas en el servidor y su base en local se actualiza para ser una copia exacta de la del repositorio en ese instante.

Si dos auditores modifican el mismo fichero antes de que el otro termine de editarlo, el sistema detectará un conflicto y guardará las dos versiones, pero obligará al próximo que abra ese documento a elegir una versión y a descartar la otra.

Estos EGAs se tienen que completar partiendo de las evidencias obtenidas de los clientes en las que se encuentra toda la información relacionada con la aplicación del ITGC. Estas evidencias son solicitadas en el documento de requerimientos de información, un documento de tipo *xlsx* donde se listan los requerimientos solicitados y se agrupan por su área de aplicación.

	A	B	C	D	E	F
1	REVISIÓN DE LOS SISTEMAS DE INFORMACIÓN					
2	Ref.	Concepto	Información adicional	Estado	Responsable	Observaciones
60	**4. Seguridad de la información**					
61	**4.1. Seguridad a nivel global**					
62	4.1.1	Políticas y procedimientos formalizados de seguridad de la información, y anexos asociados		Pendiente		
63	4.1.3	Procedimiento formalizado de gestión de usuarios (altas, bajas y modificaciones de accesos) en los sistemas de información.		Pendiente		
64	4.1.4	Listado de altas y bajas de empleados (desde RRHH) producidas en el ejercicio auditado (del 01/01/2018 al 31/12/2018).	En el que se detalle, a ser posible: nombre del empleado, departamento o posición, fecha de alta, fecha de baja, aplicaciones a las que tenía acceso, etc.	Pendiente		
65	4.1.6	Procedimiento formalizado de revisión de usuarios en los sistemas de información.		Pendiente		
66	4.1.7	Informes de revisión de acceso de los usuarios (bloqueados, inactivos, intentos de acceso no autorizados, etc.) en los sistemas de información incluidos en alcance durante el ejercicio auditado (del 01/01/2018 al 31/12/2018).		Pendiente		
67	4.1.8	Informes de revisión de los perfiles / permisos de los usuarios en los sistemas de información incluidos en alcance durante el ejercicio auditado (del 01/01/2018 al 31/12/2018).		Pendiente		
68	**4.2. Seguridad del Directorio Activo (Controlador del dominio Windows)**					
69	4.2.1	Política de contraseñas del Directorio Activo.	En la que se detalle: caducidad, longitud mínima, complejidad, histórico, intentos permitidos erróneos, tiempo de bloqueo, etc.	Pendiente		
70	4.2.2	Directivas de auditoría del Directorio Activo.		Pendiente		
71	4.2.3	Listado de usuarios del Directorio Activo.	Extracción del Directorio Activo en la que se observe: UserName, FullName, Groups, PswdCanBeChanged, PswdLastSetTime, PswdRequired, PswdExpires, PswdExpiresTime, AcctDisabled, AcctLockedOut, AcctExpiresTime, LastLogonTime, etc.	Pendiente		
72	4.2.4	Configuración de las cuentas "Administrador"/"Administrator" y "Invitado"/"Guest" del Directorio Activo.		Pendiente		
73	4.2.5	Listado de grupos de administración del Directorio Activo y listados de usuarios en estos grupos.		Pendiente		
74	**4.3. Seguridad de la aplicación QAD MFG/PRO**					
75	4.3.1	Política de contraseñas de la aplicación.	En la que se detalle: caducidad, longitud mínima, complejidad, histórico, intentos permitidos erróneos, tiempo de bloqueo, etc.	Pendiente		
76	4.3.2	Directivas de auditoría de la aplicación.		Pendiente		
77	4.3.3	Listado de usuarios con acceso a la aplicación.	Extracción de la aplicación en la que se observe: nombre de usuario, nombre completo, fecha de último acceso, caducidad, bloqueo del usuario, fecha de creación, etc.	Pendiente		

Figura 6. Ejemplo del documento enviado a los clientes con los requerimientos de información solicitados

Este sistema empezó a operar en 2008 y desde esa fecha ha ido recibiendo actualizaciones, pero la forma de trabajar con él se ha mantenido igual durante estos años. Esta forma de operar se diseñó en un momento en el que no todos los clientes tenían acceso a internet disponible para los auditores en sus oficinas, por lo que se ideó esta arquitectura para poder trabajar en sus oficinas de forma off-line.

Actualmente no se saca partido de esta arquitectura debido a que todos los clientes disponen de acceso a internet para los auditores y en ningún momento se trabaja sin conexión.

5.2.- Modelización de los procesos de una auditoría de sistemas

Para comprender mejor el contexto en el que se realiza una auditoría de sistemas es importante analizar a alto nivel cuáles son los procesos involucrados en esta y ver cual es su ciclo de vida, así se puede analizar cómo se están realizando estos procesos y abrir la posibilidad a identificar puntos de mejora que podrían ser solucionados utilizando un nuevo sistema.

Para realizar esta tarea se van a modelizar los procesos relevantes utilizando la notación BPMN (Business Process Model and Notation), una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo. Se ha optado por esta notación debido a que es una notación estándar, que es fácilmente legible y entendible para todos los stakeholders y que es ampliamente utilizada para modelizar todo tipo de procesos en todo tipo de negocios.

5.2.1.- Visión general de una auditoría informática

En este proceso se han modelizado todos los pasos correspondientes a al ciclo de vida de una auditoría informática, poniendo el foco en los pasos previos y posteriores al propio trabajo de auditoría, debido a que los pasos correspondientes a la realización del trabajo van a ser analizados en detalle más adelante (corresponden a los puntos 8 y 9 de la modelización). Estos pasos previos y posteriores incluyen las reuniones previas al inicio del trabajo, la preparación necesaria para este, la evaluación de los sistemas auditados a partir del trabajo realizado y el cierre de la auditoría. En este proceso se ven involucrados tanto el equipo financiero como el equipo de sistemas, por lo que será importante identificar las dependencias entre los dos equipos y ver cómo se comunican entre ellos. La modelización es la siguiente:

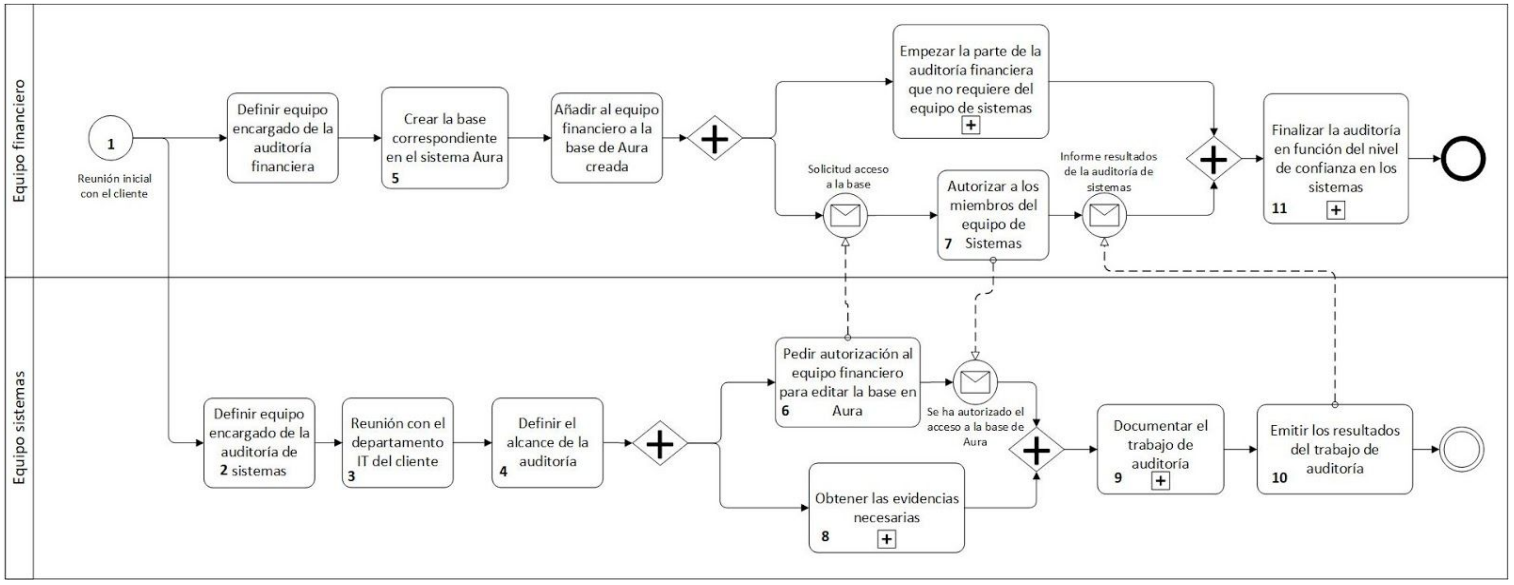


Figura 7: Modelización general de una auditoría informática

A continuación se explican en detalle los pasos más importantes representados en el diagrama de flujo anterior:

1. **Reunión con el cliente:** En esta reunión se discute principalmente de asuntos financieros y se concreta el alcance de la auditoría financiera. En principio puede parecer que no tiene ninguna influencia en el alcance de la auditoría de sistemas, pero esto realmente no es así, ya que dependiendo del alcance de la parte financiera los auditores financieros van a extraer información de distintos sistemas, y son estos sistemas los que van a entrar dentro del alcance de la auditoría informática. Así que la importancia de este paso es debido a que en esta reunión se empieza a definir cuáles serán estos sistemas.

2. **Definir equipo encargado de la auditoría de sistemas:** Antes de empezar con el trabajo es importante definir el equipo que se va a encargar de ello. Una auditoría estándar está conformada normalmente por estos 3 perfiles:

    - Gerente: Es siempre una persona con dilatada experiencia y es la responsable de gestionar el equipo, de revisar en última instancia el trabajo hecho y de evaluar la confianza en los sistemas auditados. También se encarga de las relaciones con los responsables del departamento de IT del cliente y de organizar las reuniones más importantes.

    - Auditor senior: Es un auditor con experiencia, cuyo rol principal es documentar las partes más complejas y críticas de la auditoría, manejar las relaciones con los interlocutores del cliente y revisar el trabajo realizado por el auditor junior. Su trabajo y los documentos revisados por él son luego revisado directamente por el gerente, de esta forma el trabajo a revisar por el gerente ha pasado por su filtro primero y es menos probable que contenga errores, facilitando así su revisión.

    - Auditor junior: Es auditor poco experimentado, su trabajo consiste principalmente en documentar la mayor parte de la auditoría y suele haber más de uno en el mismo equipo. Su trabajo es revisado por el senior, el cual le da *feedback* del trabajo realizado y resuelve las dudas que puedan tener.

3. **Reunión con el departamento de IT:** Esta reunión es llevada a cabo por el equipo de sistemas íntegramente, es de carácter más técnico y está enfocada principalmente a entender el entorno IT de la compañía a la que se va a auditar. En esta reunión se pregunta al cliente sobre los procedimientos y sistemas que utilizan para ellos. En estas reuniones se podria darse el caso de que el equipo de sistemas identificara un sistema que procesa datos que potencialmente van a utilizar los auditores financieros, en este caso el gerente del equipo de sistemas responsable se comunicaría con el equipo financiero para discutir sobre si es necesario o no incluir el sistema en el alcance. Esta reunión puede repetirse varias veces si desde el equipo se considera que no se tiene un conocimiento suficiente del entorno IT del cliente y se tienen dudas sobre algunos de sus procedimientos o sistemas.

4. **Definir alcance de la auditoría:** Una vez se ha realizado la reunión con el equipo de IT del cliente y se ha discutido el resultado con el equipo financiero el gerente del equipo de sistemas define el alcance que tendrá la auditoría de sistemas. Este alcance será el acordado en el punto 1 si no se ha encontrado ningún sistema que se deba de auditar en la reunión con el cliente, en caso de que se encontrara y solo si el equipo financiero lo considerara necesario se añadiría este sistema en el alcance.

5. **Crear la base correspondiente en Aura:** El líder de la auditoría contable, una persona del equipo financiero con un alto rango dentro de la empresa y miembro del Registro Oficial de Auditores de Cuentas (ROAC) es quien se encarga de crear la base asociada dentro del sistema Aura. Esta base servirá como repositorio de todas las evidencias obtenidas y toda la documentación generada para sustentar el informe de auditoría que se emitirá al terminar la auditoría contable. Solo los jefes del equipo financiero tienen privilegios para crear bases dentro de Aura, los gerentes y directores del equipo de sistemas no están autorizados.

6. **Pedir autorización para editar la base:** Desde el equipo de sistemas se debe de realizar una solicitud para poder acceder y editar los documentos de la base, esta solicitud se cursa mediante una interfaz web y en ella se debe de especificar el perfil con el que queremos acceder a la base de estos 3:

    - Lectura: Sólo se puede consultar información y no se puede modificar nada. Útil para consultar la base del año anterior para ver cómo se realizó la auditoría ese año.
    - Escritura: Se puede consultar y editar los documentos de la base, es el perfil utilizado para los auditores junior.
    - Revisión: Los mismos permisos que el rol de escritura pero además permite revisar los documentos de otros miembros del equipo. Es el perfil utilizado por los auditores senior y los gerentes.

7. **Autorizar a los miembros del equipo de sistemas:** Este paso es simple pero a veces se puede demorar en el tiempo. Esto es debido a que la solicitud cursada en el paso anterior es recibida por el autorizador via correo electronico y en el caso de recibir muchas solicitudes o una gran cantidad de correos nuestra solicitud puede que nunca llegue a ser atendida, por lo que se debe de volver a enviar o se tiene que avisar al gerente responsable de aceptarla.

8. **Obtener las evidencias de información necesarias:** Una parte vital del trabajo de auditoría consiste en la obtención de la información y de las evidencias que darán soporte a las conclusiones a las que se llegará al final de este proceso. Debido a la criticidad de este paso, su mayor complejidad y el número de interacciones entre los actores involucrados se ha decidido analizar este paso como un proceso independiente en el punto siguiente.

9. **Documentar el trabajo de auditoría:** En base a las evidencias obtenidas en el paso anterior se procede a documentar el testeo de los ITGCs que aplican a cada sistema y se concluyen los resultados para cada control en función de estas evidencias. En este paso es donde se realiza la mayor parte del trabajo de la auditoría, debido a su criticidad, su mayor complejidad y el número de interacciones entre los actores presentes en este paso se ha decidido analizarlo como un proceso independiente en el punto siguiente.

10. **Emitir los resultados del trabajo de auditoría:** Una vez se han documentado todos los controles para cada sistema en el alcance de la auditoría, el gerente responsable de esta debe emitir una conclusión en la que se decide el nivel de confianza que se puede tener en cada uno de estos sistemas. Existen 3 niveles distintos de confianza:

    - Nivel 1: El entorno de control del sistema es adecuado y se considera que no conlleva riesgo proceder con un enfoque en controles en la auditoría financiera.
    - Nivel 2: El entorno de control del sistema presenta deficiencias, pero se han realizado controles mitigantes que nos permiten proceder con un enfoque en controles en las partes del sistema en las que se ha mitigado el riesgo.
    - Nivel 3: El entorno de control del sistema presenta deficiencias y no se han podido realizar controles mitigantes para reducir el riesgo de confiar en el sistema, por lo que no se puede proceder con un enfoque en controles y se debe realizar un enfoque sustantivo en la auditoría financiera.

11. **Finalizar la auditoría en función de la confianza en los sistemas:** Tal y como se ha comentado en el punto anterior, el nivel de confianza en los sistemas es lo que permite realizar el trabajo de auditoría contable con un enfoque en controles o no, por lo que el equipo financiero necesita el trabajo del equipo de sistemas para poder tomar este enfoque, que es más rápido y menos costoso que el enfoque sustantivo.

5.2.2.- Realización de la auditoría informática

La propia realización de la auditoría consta principalmente de dos fases vistas en la modelización anterior y correspondientes al paso *8. Obtener las evidencias de información necesarias* y al paso *9. Documentar el trabajo de auditoría* respectivamente. Cada uno de estos pasos se ha analizado y modelizado como un proceso independiente.

Obtención de las evidencias de información necesarias

En este proceso se han modelizado todos los pasos necesarios para la obtención de las evidencias necesarias, esto incluye decidir los requisitos de información que deben satisfacer las evidencias que mandará el cliente, la validación de las evidencias recibidas y su posterior organización para que la documentación sea más ágil. En el proceso se ven involucrados tanto el equipo de sistemas como el departamento de IT del cliente. A continuación se adjunta la modelización del proceso:

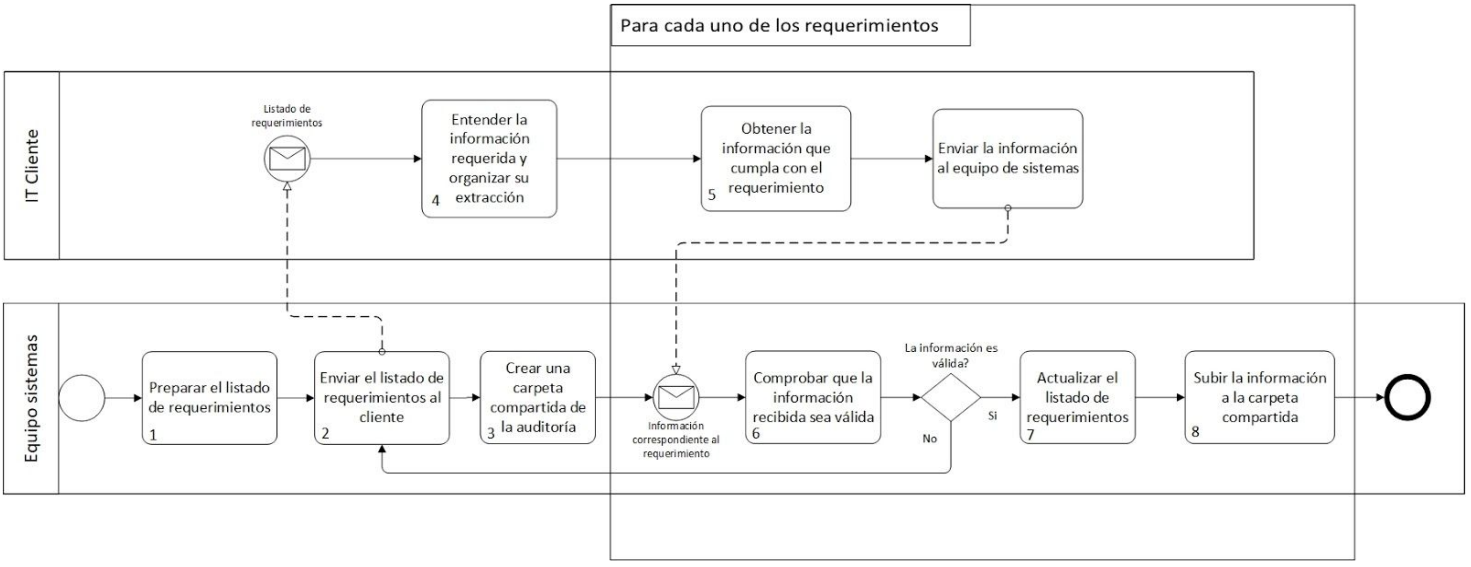


Figura 8: Modelización de la obtención de evidencias

A continuación se explican en detalle los pasos más importantes representados en el diagrama de flujo anterior, cabe destacar que los pasos 5-8 se realizan para cada uno de los requerimientos que se listan:

1. **Preparar el listado de requerimientos:** Para cada uno de los ITGCs que se van a testear y documentar es necesaria cierta información de los sistemas de los clientes que cumplan con uno o varios requisitos en concreto. Por ejemplo, si queremos testear que existe un control en la configuración de las contraseñas de los usuarios con acceso a cierta aplicación, se requiere una evidencia que contenga la configuración de la política de contraseñas de esta aplicación.

En este paso se deciden qué requerimientos de información van a ser necesarios para los controles que se auditarán en los sistemas de la compañía. Estos requerimientos suelen ser siempre los mismos para cada ITGC a testear, pero no existe ninguna plantilla con la relación entre cada control y sus requerimientos, sino que los auditores con más experiencia saben que requerimientos corresponden para los controles que se van a testear en vista a su experiencia en las auditorías previas. Estos son plasmados en un documento tipo Excel en el que se agrupan por área de aplicación y dónde el estado inicial de cada requerimiento es “Pendiente”.

2. **Enviar el listado de requerimientos al cliente:** El listado de requerimientos es filtrado por los requerimientos en estado “Pendiente” y luego es enviado, normalmente a través del mail, al departamento de IT del cliente auditado. De esta forma el cliente está informado de todos los requerimientos pendientes que se le solicitan desde el equipo de sistemas.

3. **Crear una carpeta compartida de la auditoría:** En una auditoría trabajan a la vez más de un auditor, por lo que tener en local las evidencias que se obtendrán del cliente y mandarlas a todo el equipo cada vez que se reciben de nuevas sería muy ineficiente. En esta situación lo que se hace es crear una carpeta compartida con los miembros del equipo en la herramienta Google Drive, de forma que a medida que lleguen los requerimientos estos van a ser subidos a la carpeta compartida y todo el equipo tendrá acceso a ellos.

4. **Entender la información requerida y organizar su extracción:** En el momento que el departamento de IT del cliente recibe el listado de requerimientos, este pasa a ser responsable de la correcta extracción de las evidencias que cumplan con estos requerimientos y de la asignación de esta extracción a la persona responsable (Un cliente con varios sistemas tendrá responsables distintos para cada sistema). En el caso de que tengan algún tipo de dudas en como realizar la extracción, desde el equipo de sistemas se intentará solucionar los problemas que tengan vía telefónica o desplazándose a las oficinas del cliente y reuniéndose con el equipo de IT encargado de la extracción si llegara a ser necesario.

5. **Obtener la información que cumpla con el requerimiento:** Este paso se realiza para cada requerimiento solicitado al cliente y es el responsable de la extracción quién realiza este paso obteniendo las evidencias que cumplan con el requerimiento que le ha sido asignado.

6. **Comprobar que la información recibida sea válida:** En cuanto se reciben las evidencias del cliente para cada requerimiento estas tienen que ser validadas, debido a que no se puede dar por bueno todo lo que envía el cliente sin antes asegurarse que las evidencias obtenidas cumplen con el requerimiento solicitado. En el caso de que las evidencias obtenidas cumplan con este, se procederá a actualizar el listado de requerimientos, en el caso de que las evidencias no cumplan con el requerimiento, se tiene que volver al paso 2 dónde se volverán a solicitar los requisitos que estén pendientes.

7. **Actualizar el listado de requerimientos:** Una vez las evidencias han sido validadas es muy importante actualizar el listado de requerimientos modificando el estado de los requerimientos satisfechos, que pasarán de “Pendientes” a “Recibidos”. De esta forma se mantiene en todo momento un control sobre la información de la que se dispone y la que está pendiente de recibir, facilitando así la organización de la documentación de los controles dentro del equipo. El listado se tiene que subir a la carpeta compartida siempre que se actualice y se debe trabajar siempre sobre la última versión subida, así todo el equipo tiene acceso al listado actualizado en cualquier momento.

8. **Subir la información a la carpeta compartida:** Las evidencias recibidas y validadas en los pasos anteriores se suben a la carpeta compartida creada en Google Drive en este paso. De esta forma todos los miembros del equipo pueden acceder a toda la información recibida en el momento de documentar los controles auditados.

Documentación del trabajo de auditoría

En este proceso se han modelizado todos los pasos que se siguen al documentar el testeo de los ITGCs definidos para cada sistema. Esto incluye la creación de los documentos en los que se documentaran las conclusiones de cada control, el tratamiento de las evidencias obtenidas anteriormente y el proceso de preparación y revisión de estos documentos. Este proceso es realizado de forma interna por el departamento de Sistemas. A continuación se adjunta la modelización del proceso:

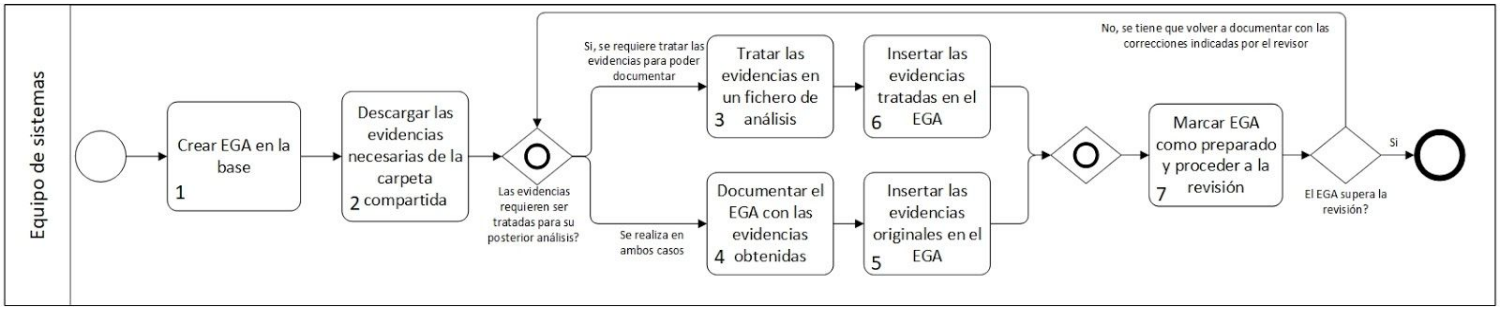


Figura 9: Modelización de la documentación de una auditoría informática

1. **Crear EGA en la base:** Para el testeo de cada uno de los ITGCs a documentar primero se crea el control que representa ese ITGC, en caso de que el control no este creado ya, y luego se crea el EGA asociado a este control, dónde se redactarán las conclusiones a las que se llegue al examinar las evidencias obtenidas relacionadas con este control y dónde se documentará todo el proceso. Estos documentos son ficheros en formato Excel, con una pestaña personalizada que permite añadir links a otros EGAs y a ficheros de varios tipos que se encuentren dentro de la misma base a la que pertenece el documento.

2. **Descargar las evidencias necesarias de la carpeta compartida:** Para la documentación del testeo del ITGC asociado al EGA creado en el paso anterior son imprescindibles las evidencias obtenidas en el proceso anterior que cumplan con los requisitos de información para el control asociado. Por lo que se accederá a la carpeta compartida en la que se subieron las evidencias obtenidas y se descargarán las que sean de utilidad para documentar la existencia y la efectividad de este control.

3. **Tratar las evidencias en un fichero de análisis:** Muchas veces el formato en el que se reciben las evidencias (ficheros de configuración en formato txt con muchos campos, resultados de distintas querys en las tablas de los sistemas en formato Excel con muchas filas y columnas...) no es el idóneo para su comprensión, por lo que es muy difícil extraer conclusiones en base a las evidencias en este formato. En estos casos lo que se hace es tratar estas evidencias y analizarlas en un fichero aparte, normalmente un fichero Excel en un formato que ayude a su comprensión, de esta forma se pueden aplicar filtros para visualizar mejor la información que contienen, facilitando la extracción de conclusiones sobre la existencia y la efectividad del ITGC que se documenta en el EGA. En el caso que no sea necesario tratar las evidencias originales este paso no se realiza.

4. **Documentar el EGA con las evidencias obtenidas:** Una vez se tienen las evidencias, con su respectivo fichero de análisis en caso de que sea necesario, es el momento de documentar la existencia y efectividad del ITGC asociado al EGA en el que se va a documentar. Al analizar las evidencias el auditor opina primero sobre la existencia del control, es decir, si este se encuentra implementado, en caso negativo se concluye que el ITGC no se encuentra implementado. En caso afirmativo se procede a opinar sobre su efectividad, es decir, si el control está bien diseñado y si opera correctamente. En ambos casos, el auditor debe de sustentar su opinión con las evidencias obtenidas anteriormente, por lo que no se puede afirmar que un ITGC se encuentra bien implementado si no se tiene la información que lo corrobore.

   Cuando se encuentran deficiencias en la implementación del control se puede proceder a establecer un control mitigante, debido a que esta auditoría sirve para dar soporte a la auditoría contable, si se encuentran evidencias de que las deficiencias en el ITGC no han afectado a los estados financieros de la compañía durante el período auditado el riesgo de que la información obtenida de estos sistemas pueda ser errónea queda mitigado y por lo tanto se puede confiar en el sistema. Aún así, se marcará el control como deficiente y se elaborará una recomendación para que el cliente sepa de esta deficiencia y proceda a subsanarla si lo considera necesario.

5. **Insertar las evidencias originales en el EGA:** Todas las evidencias obtenidas de los sistemas de los clientes y que sean utilizadas para la elaboración de conclusiones en el EGA tienen que adjuntarse en el mismo documento. De esta forma queda registrado a partir de qué información se elaboraron estas conclusiones y en caso de que hubiera una revisión externa se podría defender porqué se concluyó de esta forma partiendo de estas evidencias.

6. **Insertar las evidencias tratadas en el EGA:** En el caso de que las evidencias se hayan tratado, se tiene que subir también el fichero donde se analizan en el EGA. De esta forma se puede ver más fácilmente el proceso que ha seguido el auditor para elaborar sus conclusiones que partiendo de las evidencias originales sin tratar, y esto es de gran ayuda en el caso de que se produzca una revisión externa.

7. **Marcar el EGA como preparado y proceder a la revisión:** En cuanto el auditor considera que el EGA se encuentra finalizado procede a marcarlo como preparado, en este momento debe de informar de las deficiencias encontradas al auditar el control, en el caso que existan, o marcar que no se han encontrado excepciones en el control.

   En cuanto el EGA se encuentra preparado se asigna al responsable de la revisión, este deberá de revisar el trabajo hecho por el auditor y decidir si el EGA está correcto o se tiene que modificar algo. En caso de que el revisor considere que el EGA no es del todo correcto le puede devolver el documento al auditor y comentar con él lo que se debe modificar para que esté correcto o puede modificarlo él directamente. Cuando esta todo correcto el EGA pasa la revisión y las conclusiones sobre el ITGC que se ha documentado servirán para la emisión del informe sobre la confianza en el sistema y para elaborar el listado de recomendaciones que se trasladarán al cliente.

5.3.- Puntos fuertes y débiles del sistema actual

En este apartado se analizarán las características del sistema actual que sí son de utilidad para los auditores de sistemas, los puntos fuertes, y también las características que existen y no tienen utilidad o que directamente no se encuentran en el sistema y se considera que serían necesarias, los puntos débiles.

Puntos fuertes

- **Conexión segura con el servidor que almacena las bases:** Para replicar la información de la base en el pc del auditor o subir los cambios realizados en ella se debe estar conectado a la red de la firma, ya sea en la red de las oficinas o mediante una VPN en caso de estar fuera de ella. De esta forma la información, que en muchos casos es confidencial, se envía a través de una red segura y controlada por el equipo de IT de la firma.

- **El acceso a una base está controlado y necesita autorización:** Para tener acceso a una base es necesario que el creador de esta o alguien en el que haya delegado estos permisos autorice al usuario para que tenga dicho acceso. De esta forma solo los auditores asignados a la auditoría tienen acceso a la base.

- **Se pueden adjuntar ficheros y enlazar a otros documentos dentro del EGA:** Dentro de un mismo EGA existe la posibilidad de subir un fichero de forma que queda enlazado y puede ser abierto directamente desde el mismo documento, también se puede insertar una referencia a otro EGA, de forma que se puede navegar y abrir directamente el nuevo documento sin necesidad de explorar la base ni buscarlo. Estas funcionalidades son posibles debido a que a los ficheros de tipo Excel o Word que son utilizados al documentar se les añade una pestaña propia de Aura cuando forman parte de la base que permite estas funcionalidades.

- **Existen distintos roles dentro de la base:** Cuándo se pide acceso a una base se debe especificar también el rol que tendrá el usuario de la base. Los roles siguen una jerarquía de forma que el rol inmediatamente superior está autorizado para lo mismo que el rol inferior más otras funcionalidades añadidas.

- **Existe el concepto *Control Deficiency*:** Para indicar que existe una deficiencia mediana/grave en un control se le asocia un *Control Deficiency* donde se detalla la deficiencia encontrada, se indica si esta se encuentra en el diseño del control o en su aplicación y si existe algún control que mitigue la deficiencia.

- **Existe la vista de tareas *Dashboard*:** Esta vista permite al auditor ver todos los EGAs que tiene asignados y de los que es responsable de preparar o revisar. De esta forma se puede visualizar rápidamente los controles que cada uno tiene que documentar o revisar y no es necesario explorar la base para buscar los que se tienen asignados.

- **Existe un listado de controles:** Al asignar un control a la base se dispone de una librería con todos los controles que cubre el departamento de SPA agrupados por las áreas que cubren.

- **Existe el concepto *Coaching note*:** Este concepto representa una anotación que puede poner el usuario y que está asociada siempre a un EGA en concreto, esta anotación se puede asignar a uno mismo o a otro usuario de la base y normalmente se utiliza para anotar puntos pendientes a documentar o aspectos que se deben modificar antes de preparar el EGA.

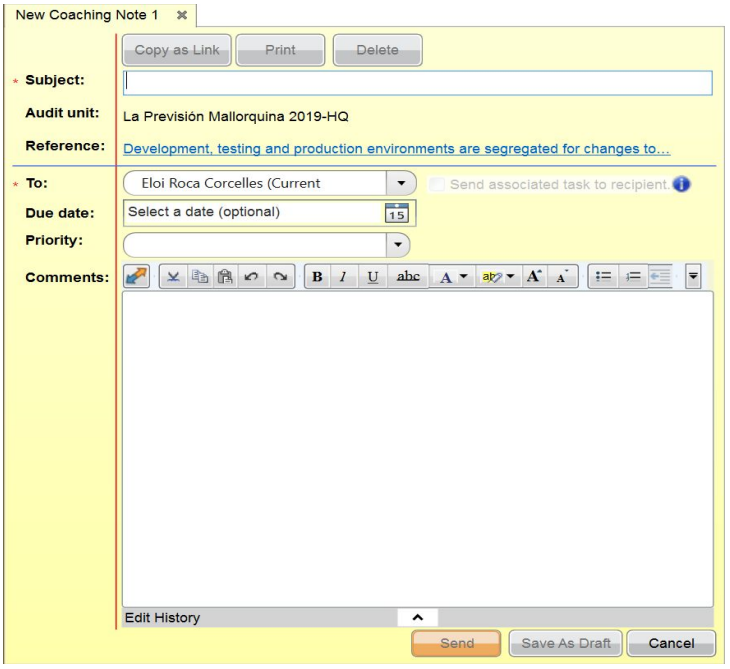


Figura 10. Ejemplo de coaching note.

Puntos débiles

- **Categorías sin relevancia.** En la base existen varias categorías pensadas estrictamente para la auditoría financiera que no sirven para nada para la auditoría de sistemas. Por ejemplo existe un apartado para el testeo substantivo, el *Substantive Testing Plan,* en la que los auditores de sistemas no realizan ningún trabajo, ya que el testeo de los ITGCs se engloban dentro del apartado de *Control Testing Plan*. Además dentro de este apartado existen muchas sub-categorías que tampoco aportan nada y todo el trabajo de sistemas cae dentro la categoría *ITGCs*, dentro de la cual solo se puede filtrar por el área en la que se aplican los controles.

  También se observa que al crear un control o un EGA se debe seleccionar el *Nature of test*, que siempre es de tipo *ITGC* por lo que no aporta nada esta distinción, y seleccionar sus *Assertions*, un concepto que sólo tiene relevancia para la auditoría financiera y que no tiene ninguna implicación para el equipo de SPA, por lo que estos dos conceptos tampoco son relevantes para la auditoría de sistemas.

Otra categoría sin relevancia son los *FLSI,* un concepto utilizado en la auditoría financiera y que agrupa los controles de un modo distinto, por lo que existe la función de filtrar por categorías o *FLSIs,* una opción que desde SPA no tiene ninguna utilidad.

- **Bases en local:** Ocupan mucho espacio en el PC del auditor (algunas bases pueden llegar a pesar más de 8Gb) y además mucho del espacio es ocupado por archivos de los auditores financieros que no tienen ninguna relevancia para el equipo de sistemas. De esta forma se da la situación que el mismo contenido de la base se encuentra replicado en todos los PCs de los auditores que participen en ella, lo cual es bastante inconveniente y da lugar a que el disco duro de los PCs con menos capacidad (256Gb) se llene rápidamente y esto obligue al usuario a perder tiempo eliminando archivos o bases antiguas para poder liberar espacio y seguir trabajando

- **Creación de ITGCs no limitada:** La forma en la que está diseñado el sistema permite crear los controles en cualquier área. Esto es debido a que para crear el control primero se tiene que seleccionar el área en la que se creará el control y luego seleccionarlo del listado de ITGCs que se testean en el departamento, pero en ningún punto se hace una comprobación de que el área seleccionada corresponda con el área que le corresponde a ese control, de forma que a veces se crean controles por error en áreas que no les corresponden.

- **Ausencia del concepto "Sistema":** Al estar el sistema diseñado para la auditoría financiera no existe ningún concepto que represente el sistema/aplicación para el cual se están testeando la aplicación de los controles generales informáticos y que permita asociarle los EGAs relacionados con este. Por este motivo si el equipo de SPA tiene que auditar varios sistemas está obligado a crear varios EGAs para el mismo control, debido a que el control es general y se aplica a todos los sistemas por igual, cambiando el título del EGA y añadiendo el nombre del sistema delante para poder diferenciarlos.

- **Ausencia de plantillas para los EGAs:** Cuando se crea un nuevo EGA este se inicializa con un Excel vacío y con sólo una pestaña personalizada. En principio rellenandola se crean las otras pestañas necesarias para documentar el control, pero a la práctica esto nunca se utiliza. Para crear la estructura del documento se utiliza el mismo EGA del año pasado o otros EGAs del mismo control, pero de clientes diferentes, y se copian las pestañas al nuevo EGA, de forma que se utiliza el documento anterior como plantilla. Esta mala práctica es llevada a cabo debido a que el propio sistema no cuenta con plantillas para los distintos EGAs que se documentan, sino que los auditores reutilizan los documentos de años anteriores o de otros clientes para utilizarlos como modelos. Esto puede conllevar a varios errores, como links rotos que apuntaban a la base del EGA anterior o el hecho de olvidarse de modificar parte de la documentación preexistente, y también puede dar lugar a que al reutilizar el mismo documento año tras año no se documente siguiendo las líneas marcadas para el período actual.

- **EGAs no limitados a su control:** La creación de un EGA en la base no está controlada, de forma que se puede crear un EGA con cualquier nombre asociado a cualquier control ya existente. Para crear los EGAs la forma de proceder consiste en copiar el nombre del control, y en caso de que el ITGC se aplique sobre un sistema concreto añadirle delante el nombre de este para poder identificarlo (en caso que sea un ITGC transversal como la existencia de un plan de contingencia no es necesario).

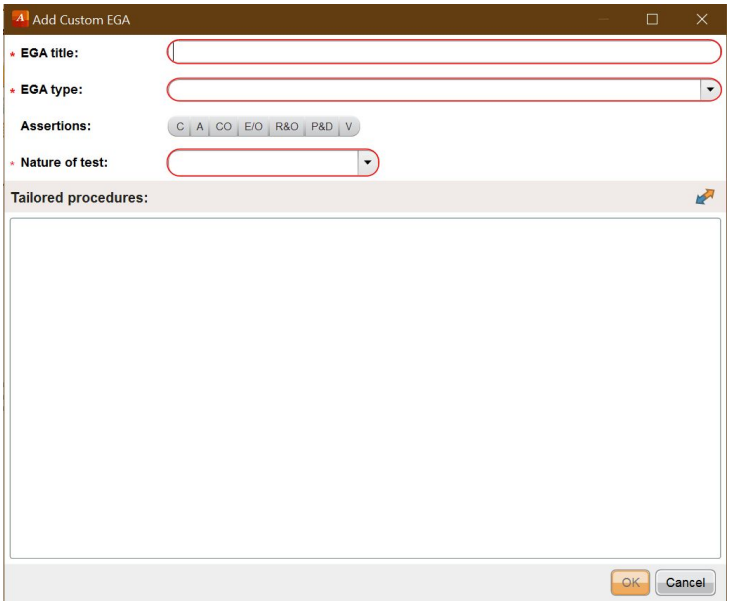


Figura 11: Ejemplo de la creación de un EGA.

- **El proceso de creación de ITGCs y EGAs en una base vacía no está automatizado y es poco intuitivo:** Cuando se empieza una nueva auditoría el área de ITGCs en el apartado de control testing está casi vacía, sólo se crea un EGA por defecto, y con todas las áreas de los controles deshabilitadas. Para poder empezar a montar la estructura de la base se tienen que seguir varios pasos: primero seleccionar las áreas que entrarán en alcance para que aparezcan habilitadas, seleccionar los riesgos asociados a cada área (siempre son los mismos y están predefinidos), proceder a crear los controles para cada pareja de área y riesgo y por último crear los EGAs asociados a cada control. Este proceso no es trivial y aunque el resultado final sea siempre el mismo no existe ninguna forma de marcar los controles que se van a realizar y que se inicialice la base automáticamente.

- **Asignación automática del revisor al crear un EGA:** Todo EGA tiene que tener asignado un responsable para su realización y un revisor como mínimo. Cuando se crea uno nuevo en la base se asigna como responsable al creador del documento, pero por defecto se asigna como revisor al propietario de la base, perteneciente al equipo de auditoría financiera y que no tiene interés alguno en tener ese EGA asignado. Manualmente se tiene que cambiar esta asignación y asignar a los miembros de SPA, y en caso de que al auditor se le olvide este paso, su EGA se asignará al propietario de la base el cual se quejara y avisará al gerente de SPA de la situación.

- **La gestión de los conflictos en los EGAs es mejorable:** Los conflictos suceden cuando dos o más auditores modifican un mismo fichero de forma paralela y luego replican contra el servidor central. El primer cambio que recibe es aceptado pero en cuanto recibe el segundo el sistema detecta un conflicto y congela el archivo, de forma que la próxima persona que abra ese archivo va a tener que seleccionar una de las dos versiones que se encuentran en la base que se convertirá en la versión definitiva y todos los otros cambios serán descartados. Esto unido a que la interfaz para solucionar estos conflictos es muy poco intuitiva puede causar errores en la solución de los conflictos haciendo que se pueda perder todo el trabajo hecho por un auditor en ese documento.

- **Falta de clasificación de los ficheros adjuntados a la base:** Los ficheros en la base se clasifican de la misma forma que los controles y los EGAs (las mismas categorías y subcategorías que siguen estos se aplican a los ficheros) con la diferencia de que los ficheros pueden quedarse sin clasificar, de forma que están referenciados por los EGAs que les apuntan pero en la librería de documentos no aparecen clasificados en ninguna categoría. Cuando adjuntamos un fichero directamente en un EGA este no se mapea en la misma categoría que la del propio EGA, sino que se añade a la librería de documentos sin categoría alguna. Esto obliga a que los ficheros que adjunta SPA se deben nombrar siguiendo una convención específica para que los auditores financieros tengan controlados los ficheros que corresponden a SPA y los suyos. Por ejemplo, si un fichero se utiliza para el testeo de un control general informático del área de seguridad se deberá nombrar “SPA_SEG_*NombreDelDocumento*” para de esta forma aclarar que es un fichero del equipo de SPA para el área de seguridad.

5.4.- Conclusiones del estudio del contexto

Una vez se ha analizado en detalle el contexto de la compañía y del departamento de SPA en concreto. Se ha modelado el ciclo de vida de una auditoría informática, analizando en el proceso cómo los usuarios interaccionan con el sistema actual durante el transcurso de esta, y una vez entendidas las fortalezas y debilidades del sistema actual, Aura, se puede concluir lo siguiente:

- Aura no cubre las necesidades específicas de las auditorías informáticas. Se observa que no existe ningún elemento en la base que represente a un sistema informático, de forma que los auditores deben sortear esta deficiencia creando múltiples EGAs para el mismo control y especificando en el nombre para que sistema tratan, pero esto implica que no existe ninguna forma de filtrar por sistema ni de organizar los controles en función de estos. Se observa que tampoco existe el concepto de ITGC y que el sistema no controla la creación de EGAs ni controles en la base, ya que para Aura no existe ningún listado con.

- Aura no proporciona plantillas para los EGAs. El sistema no ofrece ningún tipo de plantilla o modelo a seguir cuando se crea un EGA nuevo en la base, de forma que los auditores siempre tienen que coger un EGA antiguo que documente al mismo ITGC y copiarlo en el EGA actual. Esto implica que se van arrastrando EGAs de años anteriores y a veces estos EGAs no están actualizados y no están en línea con lo que se audita actualmente, de forma que los auditores con menos experiencia los siguen como guía y luego al revisarlos los auditores senior o los gerentes se dan cuenta del error y se tiene que volver a rehacer, perdiendo un tiempo que se podría haber evitado si existiese una plantilla actualizada a seguir por estos auditores junior.

- Aura tiene muchas categorías, funcionalidades y opciones irrelevantes para la auditoría informática. Esto es debido a que estas funciones son específicas para la auditoría financiera, y debido a que se realizan las dos auditorías en el mismo sistema estas opciones se encuentran presentes aunque no tengan ninguna utilidad.

- Aura es un sistema antiguo y con una tecnología obsoleta. El desarrollo de Aura tuvo lugar en 2008, desde ese momento hasta ahora se han ido recibiendo actualizaciones menores incluyendo nuevas funcionalidades y opciones, pero la arquitectura del sistema y su tecnología utilizada sigue siendo la misma. El hecho de estar obligados a tener replicadas las bases en local es innecesario actualmente y solo ralentiza el trabajo de los auditores.

Es por estos motivos que se considera necesario proveer a los auditores de este sistema con una herramienta que se adapte a sus necesidades, que ofrezca nuevas funcionalidades y que elimine las innecesarias, y que utilice una arquitectura y tecnología actual que agilice el trabajo de los auditores y les ahorre el mayor tiempo posible.

6.- Visión del proyecto y oportunidades de mejora

6.1.- Visión del proyecto

El nuevo sistema a desarrollar será utilizado por los integrantes del departamento de SPA y dará soporte a todas las tareas que se realizan en una auditoría informática, de forma que los auditores realizarán todo el trabajo de auditoría en este sistema y una vez finalizado, los *Gerentes* del departamento elaborarán las conclusiones y el listado de recomendaciones en un documento externo que proporcionarán a los auditores financieros y a los responsables de IT de los clientes auditados respectivamente. De esta forma para los auditores financieros no cambiará su modo de trabajar y no afectará a su departamento.

El objetivo es que este sistema cubra todas las necesidades y ofrezca todas las utilidades que ya cubre y ofrece el sistema actual (Aura), pero que también tenga en cuenta las necesidades específicas de una auditoría de sistemas y añada nuevas funcionalidades que le den más utilidad y faciliten el trabajo de documentación a los auditores.

De esta forma el departamento tendrá una herramienta mucho más potente para realizar su trabajo que permitirá una mejor gestión de los recursos y del tiempo empleado en cada auditoría, de forma que se perderá menos tiempo en preparar la estructura de la base y sus EGAs y se podrá invertir este tiempo en realizar el propio trabajo de auditoría. A la larga esto permitirá que se puedan abarcar más clientes, actualmente solo se realizan auditorías informáticas en los de un tamaño significativo, y ofrecerá un gran valor añadido al departamento.

6.2.- Oportunidades de mejora

Una vez clarificada la visión del proyecto, es importante identificar las oportunidades de mejora respecto a la situación actual que nuestro sistema ofrecerá. Estas mejoras van a ser experimentadas por los actores implicados en una auditoría de sistemas y sirven como motivación para la realización del proyecto.

Cubrir los puntos débiles del sistema actual

En el punto *5.3.- Puntos fuertes y débiles del sistema actual* se han identificado, analizado y tratado en detalle los puntos fuertes y débiles de Aura, el sistema actual. La especificación y el diseño del futuro sistema tiene que tener estos puntos muy presentes, manteniendo los puntos fuertes que ya cubre el sistema actual (los usuarios esperan mantener las funcionalidades que son de su agrado) y sobre todo solucionado todas las deficiencias encontradas para Aura, cubriendo así las necesidades que no son cubiertas en la actualidad. Las mejoras en concreto para este punto son las siguientes:

- **Simplificar la base:** Todas las categorías irrelevantes y que no tienen importancia para la auditoría de sistemas no se van a tener en cuenta en el nuevo sistema, por lo que la estructura de la base, sus filtros y las funcionalidades que ofrecerá serán más simples e intuitivos.

- **Introducir el concepto "Sistema":** Se utilizará este concepto para representar los sistemas que se están auditando, de forma que los EGAs estarán relacionados con el ITGC que cubren y el sistema en el que se aplica. También se introducirán filtros y vistas para este concepto.

- **Plantillas para los EGAs:** Se añadirá una funcionalidad que permitirá a los responsables del departamento crear plantillas para que los demás auditores las utilicen al documentar los EGAs, de esta forma se ahorrará tiempo y se facilitarà el trabajo de documentación.

- **Controlar la creación de Controles y EGAs:** Se controlará el proceso de creación de Controles y EGAs para que no se puedan crear en áreas que no les corresponden o en controles que no tienen relación, minimizando así la posibilidad de cometer errores.

- **Automatizar creación base:** Se ofrecerá la opción de introducir los sistemas y los controles a testear para cada sistema de forma que se genere automáticamente la estructura de la base. Esto supondrá un gran ahorro de tiempo y minimizará la posibilidad de cometer errores en el proceso.

Nuevo sistema para gestionar la obtención de evidencias

Tal y como se ha observado en el punto *5.2.- Modelización de los procesos de una auditoría de sistemas,* una de las partes más importantes de este proceso consiste en la obtención de evidencias del estado de los sistemas del cliente, debido a que a partir de estas evidencias se documentaran los ITGCs asociados, por lo que una buena gestión de estas evidencias es esencial.

Actualmente el proceso de obtención y organización de evidencias se realiza de forma manual, comunicándose por mail con el departamento de IT del cliente y recibiendo a través de ese medio la gran mayoría de evidencias (algunos clientes proporcionan sus evidencias en un USB, pero representan un porcentaje muy residual). Esta forma de proceder implica que todas las evidencias que se reciben del cliente se tienen que clasificar manualmente, actualizando el documento Excel con los puntos recibidos, y luego ser compartidas con el resto del equipo subiendolas a una carpeta compartida (ideal), enviando estas evidencias en un zip a los miembros del equipo y manteniendolas en local (acceptable) o directamente cada uno organizando la información y manteniéndola en local (mala práctica). Al final a veces se da la situación en la que no se sabe quién del equipo tiene tal evidencia, donde se almacenó esa evidencia, si se actualizó el documento con los puntos solicitados, quién tiene la última versión de este documento, en que correo se recibió una evidencia que luego nadie clasificó…

Para mejorar este punto del proceso de una auditoría informática se dotará al nuevo sistema de funcionalidades que facilitarán la obtención y la organización de estas evidencias. El objetivo es que los usuarios del departamento de IT del cliente puedan subir directamente las evidencias requeridas directamente a Aura, a través de una página web donde se mostrarán los puntos pendientes de recibir y a la que se les dará acceso temporal mientras dure la auditoría.

Asimismo los auditores podrán analizar la información que el cliente suba, clasificandola como recibida si cumple con los requerimientos solicitados, y pudiendo utilizar esta herramienta como carpeta compartida donde subir ficheros de análisis y desde la cual poder linkar estos ficheros en los EGAs. De esta forma la información se encontraría siempre centralizada en la base y se reduciría la cantidad tratada en local por cada auditor, logrando así una mejor organización y un ahorro de tiempo en la gestión de estas evidencias y también agilizando la comunicación entre SPA y el departamento de IT del cliente.

Modernizar la tecnología utilizada

El sistema actual, Aura, tiene ya bastantes años y utiliza una arquitectura en la que todo el trabajo se ejecuta en local en el PC del auditor, conectándose solo al servidor principal para sincronizar la información de la base, de forma que se ocupa una gran cantidad de memoria física de estos PCs y el programa tarda un tiempo al iniciarlo en funcionar con rapidez.

Esta arquitectura se implementó debido a que cuando se implementó el sistema no todos los clientes auditados tenían conexión a Internet disponible para los auditores, por lo que se ideó esta arquitectura para que se pudiera trabajar sin conexión en sus oficinas. Actualmente, todos los clientes disponen de una conexión estable a Internet, por lo que mantener esta arquitectura física cuando su necesidad es inexistente no sería lógico.

Es por eso que el nuevo sistema utilizará las últimas tecnologías y trasladará el trabajo realizado en local en el PC de los auditores a la nube, de forma que estos se conectaran a través de un cliente web y podrán documentar directamente en la base sin necesidad de mantener una copia en local ni de ir replicando cada vez que realicen una modificación. También permitirá consultar auditorías pasadas sin necesidad de descargas bases enteras. Esto permitirá que el trabajo de documentación sea mucho más ágil y que no se pierda tiempo descargando/replicando bases ni esperando que el programa se inicialice para poder abrir una base.

7.- Partes interesadas

En este apartado se identificarán las principales partes interesadas en el desarrollo del nuevo sistema para la documentación de auditorías informáticas.

Para cada parte interesada o *stakeholder,* se identificará el interés que tiene en el sistema, el rol que tendrá durante su desarrollo y los objetivos que quiere cumplir con el sistema operativo. Estos *stakeholders* se clasificarán en 4 tipos en función del interés de cada parte.

Tema

En este punto se tratarán a los *stakeholders* relacionados con el dominio del sistema, es decir, las partes relacionadas con la realización de las auditorías informáticas por parte del departamento de SPA.

- **Firma de auditoría:** La propia compañía que factura por las auditorías que se realizan y que obtiene un beneficio económico con el trabajo de los auditores.
  Rol:
    - Dar soporte al desarrollo del sistema y proporcionar el presupuesto necesario para su realización.

  Objetivos:
    - Optimizar el tiempo invertido en las auditorías de sistemas.
    - Poder realizar más auditorías de sistemas, de forma que los auditores financieros se vean beneficiadas al tener que realizar menos trabajo.
    - Poder ofrecer este servicio a más clientes y cubrir más sistemas en el alcance.

- **Empresas auditadas:** Las propias empresas que contratan a la firma para que las audite y que esperan recibir el informe de auditoría asociado sin que se produzcan sobrecostes y antes de la fecha máxima acordada.
  Rol:
    - *No desempeñará ningún rol en el desarrollo del sistema*

  Objetivos:
    - Reducir la duración de las auditorías
    - Aumentar el alcance de las auditorías de sistemas.

- **Departamento de auditoría financiera:** El departamento encargado de la realización de toda la parte contable de la auditorías. Necesitan el trabajo y los informes de confianza en los sistemas de SPA para justificar ciertos métodos utilizados y procedimientos seguidos en el transcurso de la auditoría contable.
  Rol:
  - *No desempeñará ningún rol en el desarrollo del sistema*

  Objetivos:
  - Obtener los resultados de las auditorías de sistemas más rápidamente.
  - Disponer del trabajo del equipo de SPA para más auditorías.

- **Dirección del departamento de SPA:** Formado por los socios y directores responsables de la división de SPA. Se encarga de gestionar el control interno del departamento y negociar las asignaciones de horas para cada auditoría de sistemas.
  Rol:
  - Promover el desarrollo del sistema a la dirección de la firma

  Objetivos:
  - Optimizar el rendimiento del departamento de SPA
  - Conseguir dar servicio a más clientes sin aumentar la carga horaria del equipo

Tecnología

En este punto trataremos a los *stakeholders* interesados en la parte tecnológica del sistema, es decir, las partes que tienen algún interés en la tecnología que utilizará el sistema o en la infraestructura tecnológica que le dará soporte una vez esté operativo.

- **Proveedor *cloud:*** Se encargará de dar soporte y mantener el servidor web en el que se hospedará el sistema y contra el que se conectarán los auditores desde sus dispositivos.
  Rol:
  - *No desempeñará ningún rol en el desarrollo del sistema*

  Objetivos:
  - Hospedar el sistema en sus servidores y ofrecer un servicio que cumpla las demandas de la firma.
  - Obtener un beneficio económico con el servicio ofrecido.

- **Equipo IT:** Se encarga de dar soporte a todas las aplicaciones de la firma y también funciona como *helpdesk* para solucionar los problemas que puedan tener los trabajadores para acceder a estas aplicaciones.
  Rol:
  - Detallar como funciona la infraestructura de la firma y facilitar la integración de la aplicación en esta.

  Objetivos:
  - Obtener una aplicación segura que se pueda desplegar en la infraestructura de la firma sin comprometer su seguridad.

Uso

Este punto engloba a los *stakeholders* que tienen un interés directo en el sistema debido a que serán los futuros usuarios de este. Como usuarios finales del sistema, estos *stakeholders* tienen un gran peso y se deben tener muy en cuenta.

- **Equipo de SPA:** Los principales usuarios del sistema, los auditores del equipo de SPA, estos utilizaran el sistema para documentar el estado de los ITGCs definidos para los sistemas auditados y para consultar esta documentación al elaborar el informe de confianza en los sistemas y el listado de recomendaciones a seguir por el cliente para solventar las deficiencias encontradas en estos controles.
  Rol
  - Participar activamente en la especificación de requisitos para asegurar que el sistema especificado cubre con las necesidades de los auditores.

  Objetivos:
  - Obtener un sistema que agilice las documentación de los controles, que facilite la gestión de las bases y que aporte funcionalidades que cubran las necesidades específicas de la auditoría de sistemas.
  - Obtener un sistema rápido y intuitivo de utilizar, que no ocupe espacio en el disco local de los equipos de los auditores y que no genere conflictos al trabajar varias personas de forma simultánea.

- **Departamento IT clientes:** Los responsables del departamento de IT del cliente encargados de proporcionar las evidencias solicitadas de la aplicación de los Controles Generales Informáticos, estos van a utilizar una pequeña parte del sistema para subir estas evidencias directamente y controlar que información tienen pendiente y que puntos se dan ya por cerrados.
  Rol:
  - Informar sobre las funcionalidades que esperarían poder utilizar, estas se tendrán en cuenta al especificar y diseñar esta parte del sistema.

  Objetivos:
  - Obtener un sistema que les permita subir de forma segura todas las evidencias solicitadas y sin restricciones en los tamaños de los archivos.
  - Obtener un sistema que les permita consultar el estado de los requerimientos para los que tienen que subir las evidencias, de forma que en cada momento puedan saber que requerimientos han sido ya satisfechos y para cuales aún se deben proporcionar evidencias.

Desarrollo

En este punto se tratan los *stakeholders* interesados o involucrados en el proceso de desarrollo del sistema y no en el sistema una vez esté desarrollado e implementado.

- **Gestión de proyectos de la firma:** El equipo encargado de gestionar los proyectos impulsados por parte de la compañía, se encargan de controlar sus avances y gestionar su implementación dentro de la firma.
  Rol:
  - Comunicarse con el equipo de desarrollo, controlar el avance del desarrollo, y asegurar que todo avanza acorde a la planificación.
  - Gestionar la implementación del sistema dentro de la firma.

  Objetivos:
  - Conseguir que el desarrollo se produzca acorde con la planificación y que no se produzcan desviaciones respecto al importe presupuestado.

- **Ingeniero de requisitos:** Se encargará de estudiar y analizar el contexto del sistema y las necesidades que se pretenden solventar con la implementación de este. También se encargará de analizar y especificar los requisitos que deberá cumplir el sistema para satisfacer a sus futuros usuarios.
  Rol:
  - Realizar la especificación de requisitos

  Objetivos:
  - Identificar correctamente todos los requisitos necesarios a cumplir.
  - Especificar el sistema de una forma óptima para que el equipo encargado de su diseño entienda claramente qué sistema se debe diseñar y programar.

- **Arquitecto de software:** Se encargará de decidir la arquitectura del sistema y diseñarlo para que cumpla con los todos requisitos especificados por el ingeniero de requisitos, teniendo también en cuenta la seguridad, eficiencia y cambiabilidad del sistema durante su diseño.
  Rol:
  - Definir la arquitectura del sistema y diseñarlo teniendo en cuenta los requisitos que debe cumplir.

  Objetivos:
  - Diseñar un sistema que cumpla con los requisitos funcionales y no funcionales definidos por el ingeniero de requisitos.
  - Realizar un diseño robusto y de máxima calidad.

- **Equipo de programación:** Equipo encargado de implementar el código del sistema siguiendo la arquitectura y el diseño definidos por el arquitecto de software. Estará compuesto por varios programadores liderados por un ingeniero de software con experiencia que validará la implementación del sistema.
  Rol:
    - Implementar el código del sistema siguiendo el diseño del arquitecto de software.

  Objetivos:
    - Programar el código del sistema de forma que no contenga errores ni vulnerabilidades.
    - Testear que el sistema se comporte acorde a lo especificado y que cumpla correctamente con los requisitos especificados.
    - Entregar el código desarrollado antes de la fecha límite acordada.

- **Mantenimiento del sistema:** Una vez el sistema ya esté desarrollado y operativo, este equipo se encargará de solucionar las posibles incidencias o errores que puedan afectar a sus usuarios finales.
  Rol:
    - Solucionar las incidencias de los usuarios una vez el sistema esté operativo.

  Objetivos:
    - Solucionar las incidencias reportadas en el mínimo de tiempo posible.
    - Procurar que el sistema se encuentre siempre operativo para que los auditores puedan trabajar sin complicaciones.

Especificación del sistema

Una vez se ha analizado el contexto, entendiendo las necesidades específicas que debe cubrir el sistema a desarrollar y los objetivos específicos que pretende conseguir cada uno de los stakeholders involucrados en éste, es el momento de especificar en qué consiste el sistema propuesto en este proyecto. Esto incluye definir qué tipo de sistema se quiere desarrollar y con qué arquitectura física, establecer los conceptos que formarán el dominio del sistema, definir qué funcionalidades va a ofrecer y quienes tendrán acceso a estas funcionalidades.

8.- Visión general

El sistema propuesto para la realización de auditorías de sistemas de información y que sustituirá a Aura, el sistema actual, se analizará en este apartado, de esta forma se entenderá mejor en qué consistirá el sistema, que funcionalidades tendrá y cómo los usuarios las utilizarán.

A continuación, se definirá con detalle la arquitectura física del sistema y se comentarán las principales funcionalidades de las que constará el sistema y cómo las utilizarán los usuarios.

8.1.- Arquitectura física

El sistema propuesto pretende solucionar uno de los principales problemas arquitectónicos de Aura, que consiste en el hecho de que toda la información generada durante una auditoría se almacena en una base que debe ser forzosamente replicada en local en el PC de cada uno de los auditores participantes en ella, de forma que toda la información está siempre duplicada en cada uno de estos PCs y en un servidor central que coordina todas las modificaciones realizadas en la base. Esto se traduce en una gran ineficiencia, debido a que siempre se debe de invertir un tiempo a replicar estas bases antes de empezar a trabajar en ellas y después de modificar cualquier elemento, ya que se tiene que replicar esta modificación al servidor central para que se propague a los otros PCs.

La arquitectura física propuesta para este sistema consiste de una servidor web que se hospedará y ejecutará en la nube, al que se conectarán los auditores mediante un navegador compatible y que sólo será accesible para los auditores desde dentro de la red dela firma. En caso de que se encuentren en las oficinas de un cliente se podrán conectar al sistema utilizando una VPN, una tecnología que permite una extensión segura de la red de área local sobre una red pública o no controlada, permitiendo así que el ordenador envíe y reciba datos como si estuviera dentro la red privada con toda la funcionalidad, seguridad y políticas de gestión de esta [4].

Al estar el sistema dentro de la red privada de la compañía, este tendrá acceso también al Servicio de Directorio (SD) de la compañía, donde se encuentran registrados los usuarios de esta y mediante el cual se identifican en los servicios de la firma. Debido a que los auditores tienen sus credenciales almacenadas en su PC, el sistema solicitará estas credenciales y las validará contra el SD de la compañía, en caso de que correspondan a un usuario del sistema y sean válidas, este se conectará directamente sin necesidad de introducir la contraseña otra vez.

El sistema accederá a una BD que almacenará las bases de cada auditoría, de forma que no se deberá descargar la base en local para trabajar sobre ella, sino que se accederá online y se trabajará directamente sobre la base en el servidor.

Los responsables del departamento IT de los clientes también se van a poder conectar al sistema para subir las evidencias que cumplan con los requisitos de información que se les solicitarán. En este caso, se les crearán usuarios temporales que solo tendrán acceso a una parte muy limitada del sistema, las únicas funcionalidades a las que tendrán acceso consistirán en consultar el estado de los requerimientos y subir las evidencias solicitadas para cada punto. Por practicidad, y teniendo en cuenta su acceso tan limitado, no será necesario que accedan mediante una VPN.

A continuación se adjunta un diagrama de la arquitectura física propuesta para el sistema.

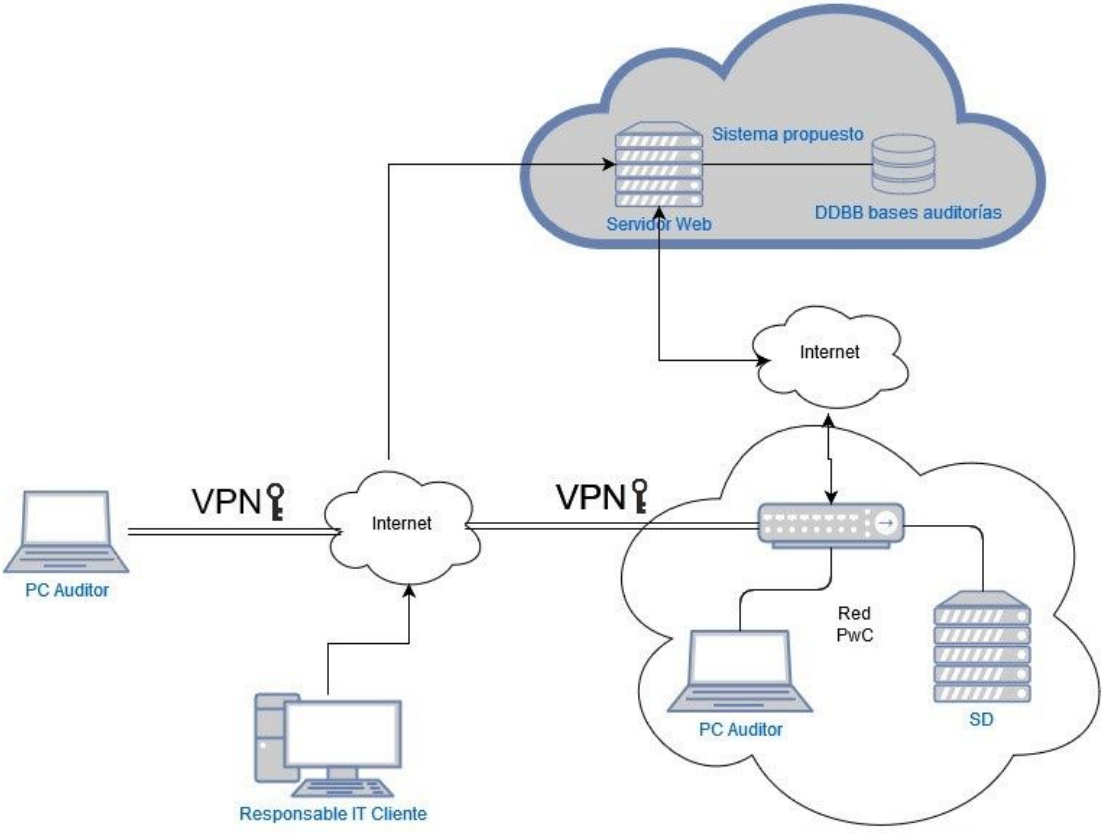


Figura 12. Arquitectura física del sistema

8.2.- Funcionalidades del sistema

El sistema será utilizado principalmente por los auditores de SPA, que lo utilizarán para gestionar y documentar todo el trabajo de auditoría que realicen sobre los sistemas informáticos de los clientes a los que auditen. Existirá también un módulo para que los clientes interaccionen con el sistema para proporcionar las evidencias solicitadas, pero sus funcionalidades serán muy limitadas. En este apartado se definirán a alto nivel las funcionalidades del sistema a las que tendrán acceso sus usuarios.

8.2.1.- Funcionalidades de los auditores de sistemas

Los auditores de sistemas del departamento, como principales usuarios del sistema, van a ser los actores con más funcionalidades disponibles dentro del sistema, cabe destacar que dependiendo del perfil y el rol de cada auditor las funcionalidades a las que tendrá acceso serán distintas.

Existirá el perfil *Manager,* reservado a los gerentes de cada oficina, que permitirá administrar y gestionar el sistema, y el perfil *Auditor*, que sólo permitirá participar en las auditorías creadas por los gerentes. Para cada auditoría, el gerente responsable de cada base asignará un rol a cada auditor del que dependerán las funcionalidades a las que tenga acceso. El propio gerente tendrá el rol de *Team Leader,* se asignará el rol de *Reviewer* a los auditores con experiencia y se asignará el rol de *Team member* a los auditores con menos experiencia y responsabilidades.

Estos 3 roles formarán una jerarquía, de forma que el *Team Member* tendrá acceso a las funcionalidades básicas de todos los auditores, los *Reviewers* tendrán acceso a estas funcionalidades más las funcionalidades específicas asociadas a su rol y los *Team Leaders* tendrás acceso a todas las funcionalidades de las que disponen los *Reviewers* más las funcionalidades específicas asociadas a su rol.

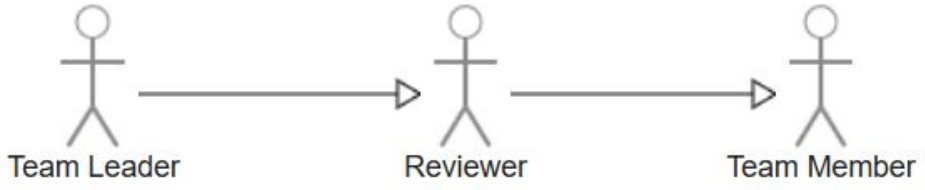


Figura 13. Jerarquía de roles en una base, el sentido de la flecha indica que el ese rol tiene acceso a las funcionalidades del rol al que apunta.

Las principales funcionalidades con las que van a contar son las siguientes:

1. **Gestión de sistemas estándares:** Funcionalidad asociada al perfil *Manager.* Permitirá a los gerentes registrar aquellos sistemas software considerados estándares, por ejemplo los ERPs SAP o Navision o las bases de datos Oracle.

2. **Gestión de ITGC estándares:** Funcionalidad asociada al perfil *Manager.* Permitirá a los gerentes definir los ITGCs considerados estándares y para cada uno de ellos decidir si su aplicación es transversal, en caso de que apliquen a todos los sistemas de la compañía (por ejemplo, el control "La sociedad dispone de un plan de contingencias informático formalizado y realiza pruebas de su aplicación" del área de Operaciones), o específicos de cada sistema, en caso de que su aplicación sea individual para cada uno de estos (por ejemplo, el control "Los cambios a programa son debidamente testeados y autorizados antes de su pase a producción" del área de Gestión de Cambios).

3. **Gestión de plantillas:** Funcionalidad asociada al perfil *Manager.* Permitirá crear, seleccionar el control para el que aplique y editar de estructura de las plantillas para la documentación de los ITGC, de forma que la estructura de cada EGA en el que se documente un control parta de la estructura de una plantilla elaborada para este. Se podrán registrar también plantillas específicas de un sistema, por lo que un control tendrá siempre una plantilla genérica y podrá tener varias de específicas. Esta mencionada estructura podrá estar en formato Excel como en la actualidad o en un formato interno del sistema.

4. **Edición de plantillas:** Funcionalidad asociada al perfil Manager. Permitirá modificar la estructura de una plantilla añadiendo, modificando y/o eliminando sus elementos. En caso de que la estructura elegida esté en formato Excel esta funcionalidad será realizada por el mismo programa de Microsoft Office y en caso de que el formato sea interno del sistema será realizado por el propio sistema. Los elementos de los que constará una plantilla serán: hojas, apartados, comentarios y tablas. Estos elementos podrán contener otros elementos siguiendo la siguiente relación: las hojas serán el contenedor de todos los demás elementos, cada hoja podrá tener varios apartados y cada apartado podrá contener comentarios, tablas y otros apartados (que aparecerán como subapartados). Los comentarios y las tablas sólo podrán contener texto.

5. **Gestión de requerimientos de información estándares:** Funcionalidad asociada al perfil *Manager.* Permitirá registrar y modificar requerimientos de información estándares, estarán asociados siempre a un área de aplicación concreta y opcionalmente a un sistema específico (por ejemplo, un requerimiento en el que se soliciten varias tablas concretas de SAP)

6. **Gestión de clientes:** Funcionalidad asociada al perfil *Manager.* Permitirá registrar y modificar la información de los clientes que se auditarán, para de esta forma poder crear las bases correspondientes a sus auditorías y registrar su personal de IT encargado de proporcionar las evidencias de los sistemas.

7. **Administración de bases:** Funcionalidad asociada al perfil *Manager.* Permitirá crear bases de auditorías, buscar bases en el sistema y cerrar una base, de forma que quedará bloqueada y no se permitirá editar ningún contenido, sólo consultar la información en modo lectura.

8. **Gestión de auditores:** Funcionalidad asociada al perfil *Manager.* Esta funcionalidad permitirá asignar un auditor a una base concreta y seleccionar el rol del auditor en la base, en el momento de asignarlos a los auditores se podrá filtrar por su nombre y por su categoría en el departamento. También permitirá modificar el rol dado a un auditor en esa base y por último permitirá des-asignar a un auditor de la base, de forma que el auditor no podrá acceder a esa base.

9. **Gestión de sistemas de clientes:** Funcionalidad asociada al perfil *Manager.* Permitirá registrar los sistemas informáticos de un cliente que van a ser auditados y/o modificar la información de los sistemas ya registrados. Permitirá seleccionar si un sistema está en alcance o no, de forma que los sistemas en alcance se podrán relacionar con los ITGCs dentro de las bases que estén activas, también permitirá clasificarlos como estándar en caso de que se correspondan a alguno de estos.

10. **Gestionar el alcance de la base:** Funcionalidad asociada al rol *Team Leader.* Permitirá seleccionar los ITGC estándares en alcance por cada área de aplicación. En caso de que el control sea *específico de sistema* se podrán asignar los sistemas para los que se testeará su efectividad. Esta funcionalidad generará automáticamente todas las áreas e ITGCs, creando un ITGC de tipo transversal para cada ITGC estándar en alcance y un ITGC de sistema para cada sistema en el que se aplique cada ITGC estándar. Esta funcionalidad dará forma a la estructura de la base en la que sólo faltará crear los EGAs.

11. **Cerrar EGAs:** Funcionalidad asociada al rol *Team Leader.* Permitirá cerrar el EGA correspondiente a un ITGC una vez lo haya revisado el *Team Leader*, de forma que el documento quedará bloqueado y no se podrá modificar. Solo el *Team Leader* tendrá la opción de volver a abrir el EGA.

12. **Administración de responsables de IT de los clientes:** Funcionalidad asociada al rol *Reviewer.* Permitirá dar de alta a los responsables de IT de los clientes al sistema. Se utilizará su mail como identificador y se inicializará con una contraseña aleatoria que el cliente podrá cambiar utilizando un link para “recuperar” esa contraseña.

13. **Gestión de requerimientos de información:** Funcionalidad asociada al rol *Reviewer.* Permitirá definir el listado de requerimientos de información que deberá cumplimentar el equipo de IT del cliente para poder testear la efectividad de los ITGCs que aplican sobre sus sistemas informáticos. Este listado se generará automáticamente a partir de los requerimientos estándares definidos por los gerentes en función de los controles y los sistemas específicos en alcance, pero podrá ser modificado manualmente añadiendo, eliminando o editando requerimientos, para adaptarse a las necesidades concretas de cada cliente y cada sistema.

14. **Asignación de requerimientos a sus responsables:** Funcionalidad asociada al rol *Reviewer.* Permitirá asignar a cada responsable de IT del cliente los requerimientos de información solicitados que le correspondan (se podrá asignar un mismo punto a varios responsables) de forma que se podrá controlar quién es responsable de cada punto faltante y los responsables sabrán en todo momento qué evidencias se les solicita a cada uno de ellos.

15. **Gestión de recordatorios:** Funcionalidad asociada al rol *Reviewer.* Permitirá definir una política de recordatorios, que son mensajes que se enviarán al mail de los responsables de IT de los clientes registrados para recordarles los requerimientos de información que tienen pendientes. En esta política se definirá la frecuencia de estos recordatorios y condiciones específicas para enviarlos (por ejemplo, solo enviar en caso de que no se haya subido ninguna evidencia en durante una setmana). También permitirá enviar recordatorios de forma manual, pudiendo editar el mensaje del recordatorio y enviándolo en cuanto se considere necesario independientemente de la política.

16. **Revisión de EGAs:** Funcionalidad asociada al rol *Reviewer.* Permitirá acceder a la vista de *Revisión,* en la que se encontrarán los EGAs que se van a tener que revisar. En cuanto el auditor asignado al EGA termine su trabajo y el EGA pase al estado *Preparado* estará listo para el *Reviewer.* Una vez terminado, este pasará al estado *Revisado* y se asignará al siguiente auditor de la jerarquía, el *Team Leader*, que al terminar su revisión cambiará su estado a *Cerrado.*

17. **Acceso al sistema:** Esta funcionalidad permitirá a los auditores acceder al sistema mediante el servicio de directorio (SD) de la compañía. Al estar estas credenciales almacenadas en el PC del auditor, el sistema las solicitará y comprobará que sean válidas con una solicitud contra el SD de la firma. En caso de que las credenciales sean válidas y el usuario esté registrado en el sistema, este accederá directamente sin necesidad de introducir ninguna contraseña.

18. **Acceso a las bases:** Esta funcionalidad permitirá acceder a las bases a las que un auditor está asignado y de esta forma acceder a las funcionalidades específicas para los elementos de la base.

19. **Clasificación de las evidencias recibidas:** Esta funcionalidad permitirá modificar y consultar el estado de los requerimientos de información asignados a los responsables del cliente. El estado de los requerimientos se inicializará a *Pendiente* y se modificará automáticamente a *En evaluación* en cuanto se suban sus evidencias correspondientes al sistema. El auditor luego podrá modificar el estado del punto como *Recibido* si son correctas o volver al estado *Pendiente de recibir* en el caso contrario. Permitirá añadir una aclaración en el requerimiento para que el responsable entienda porqué las evidencias subidas, o parte de ellas, no sirven y qué se necesita recibir en concreto. También se podrá filtrar por estado (*Pendiente recibir*, *Pendiente evaluar* y *Recibido),* obteniendo así una visión completa de los puntos obtenidos y los que aún se deben de recibir.

20. **Administración manual de evidencias:** Esta funcionalidad permitirá añadir y eliminar las evidencias que correspondan a un requerimiento de información concreto de forma manual. Esta funcionalidad es necesaria debido a que si algún cliente no se encuentra cómodo utilizando el sistema y prefiere seguir enviando las evidencias con los métodos actuales, se debe poder subir estas evidencias al sistema y seguir trabajando sin complicación.

21. **Administración de ficheros tratados:** Esta funcionalidad permitirá añadir y eliminar los ficheros de análisis que se obtengan a partir del tratamiento de las evidencias obtenidas. Estos documentos tendrán una categoría especial dentro de la base y no estarán vinculados a ningún requerimiento del cliente, pero si estarán asociados con el área de aplicación correspondiente a los controles para los que se utilice.

22. **Recibir notificaciones:** Esta funcionalidad permitirá al sistema enviar notificaciones, en formato mail, al correo de los auditores en cuanto se suban evidencias para una base en la que estén asignados. Estas alertas serán de carácter diario, de forma que si se suben varias evidencias en un día se recibirá un solo mail notificando que existen nuevas evidencias pendientes de evaluar.

23. **Administración de EGAs:** Esta funcionalidad permitirá crear EGAs a partir de las plantillas predefinidas por los gerentes. El auditor deberá seleccionar el control para el que se creará el EGA y podrá elegir entre la plantilla estándar para ese control o la plantilla específica para ese control aplicado en un sistema concreto (por ejemplo SAP) en el caso que el control sea específico para un sistema. También permitirá eliminar un EGA concreto de la base o restablecerlo utilizando una nueva plantilla.

24. **Definición de la cadena de asignación:** Esta funcionalidad permitirá seleccionar los auditores que formarán parte de la cadena de asignación del EGA. Por defecto, el usuario que cree el EGA estará en la posición más baja de la jerarquía, un auditor con el rol *Reviewer* de la base estará asignado como revisor de su trabajo y por encima de este se asignará al auditor con el rol *Owner* de la base como revisor final del trabajo. Esta cadena se podrá modificar manualmente para adaptarse a las necesidades de los auditores, por ejemplo en caso de que el *Reviewer* asignado por defecto no corresponda al revisor real.

25. **Edición de EGAs:** Esta funcionalidad permitirá completar los EGAs, cuya estructura vendrá predefinida por la plantilla, con la información concreta para los sistemas de cada cliente obtenida a partir de las evidencias de información proporcionadas. Para completar los EGAs se podrán añadir, modificar y/o eliminar hojas, títulos, comentarios, apartados, tablas, imágenes y links a otros EGAs o a documentos añadidos a la base. Para enlazar a otros EGAs y a documentos se podrá buscar el ítem en concreto que se quiera enlazar mediante la funcionalidad *Búsqueda por condiciones.* El formato de los EGAs dependerá del elegido para las plantillas, independientemente del formato elegido, esté soportará los elementos comentados anteriormente.

26. **Bloqueo de EGAs:** Esta funcionalidad servirá para bloquear el acceso a la edición de un EGA si existe un auditor que lo esté editando en ese preciso instante. En cuanto un auditor acceda a un EGA, este se bloqueará y solo permitirá que este auditor realice modificaciones, los otros auditores podrán acceder al EGA pero solo en modo lectura, de esta forma se evitarán los conflictos en la edición de EGAs.

27. **Preparar EGAs:** Esta funcionalidad permitirá marcar el EGA como *preparado* una vez el auditor haya terminado su trabajo en este. Esto implicará que el estado de este EGA cambiará de *En progreso* a *Preparado* y será asignado al siguiente auditor en su jerarquía.

28. **Filtrar por condiciones:** Esta funcionalidad permitirá filtrar los listados de EGAs, ITGCs, Evidencias o Requerimientos de información de la base en función de su nombre, Área de aplicación, Sistema asociado y/o ITGC concreto asociado. Esto ofrecerá distintas vistas a los auditores (por ejemplo, ver todos los controles y sus EGAs para un determinado sistema) de forma que se agilizará el trabajo del auditor, al no tener que buscar manualmente entre otros ítems que no interesen en ese momento. También permitirá elegir entre dos vistas para los ITGCs que dependerán del orden en el que se agrupen los elementos. La primera vista agrupará primero a los controles por su área de aplicación, después por su ITGC genérico y por último los distinguirá por su sistema o por si son transversales y la segunda vista agrupará los ITGCs por su sistema o (se agruparán en “transversales” en caso de que no tengan), después por su área de aplicación y por último los distinguirá en función del ITGC genérico que representen.

29. **Consultar el *dashboard*:** Esta funcionalidad permitirá consultar el listado de EGAs y CN que tengan asignados en cada momento, pudiendo consultar también los que estén asignados a otros auditores del equipo filtrando por auditor. Desde esta vista se podrá consultar también el estado de cada uno de los EGAs, si presentan deficiencias y el listado de documentos *linkados* en cada uno de ellos.

30. **Administración de *Coaching Notes*:** Esta funcionalidad permitirá crear, editar y eliminar *Coaching Notes,* o *CNs,* (Anotaciones asociadas a un EGA) para cada EGA de la base. Al crear una *CN* se requerirá seleccionar que auditor de la base es su receptor (se puede asignar a uno mismo para utilizar las CNs como recordatorios), se deberá añadir un título para la CN y se podrá detallar en el cuerpo de la nota el contenido que se quiera anotar.

31. **Administración de *Control Deficiencies:*** Esta funcionalidad permitirá crear, editar y eliminar (en caso de un auditor con rol *Team Member* sólo podrá eliminar los que él mismo haya creado) *Control Deficiencies,* o *CDs,* asociados a un ITGC de la base. Este elemento será creado en caso de que se detecten una deficiencias graves para el ITGC testeado en el EGA, en él se deberá describir en qué consiste la deficiencia detectada, si existe algún control compensatorio y qué riesgo supone para la auditoría.

8.2.2.- Funcionalidades de los responsables IT del cliente

Los responsables de IT designados por el cliente para proporcionar las evidencias relacionadas con la aplicación de los ITGCs van a disponer de un acceso muy limitado al sistema para proporcionar a los auditores estas evidencias. Las únicas funcionalidades de las que dispondrán serán las siguientes:

32. **Acceso al sistema:** Los responsables registrados recibirán un mail con un link que les redirigirá a una interfaz del sistema donde deberán identificarse mediante su dirección de email y contraseña. Cuando sean registrados en el sistema, se les enviará un link en un email para “recuperar” su contraseña, de forma que podrán introducir la contraseña que deseen.

33. **Consultar requerimientos de información:** Esta funcionalidad permitirá consultar el estado de los requerimientos de información (*Pendiente de recibir*, *Pendiente de evaluar* o *Recibido*) que los auditores vayan actualizando durante el transcurso de la auditoría. Permitirá filtrar por usuarios para ver los requerimientos que tiene cada responsable asignado y su estado, también ofrecerá una vista general de todos los requerimientos para que los responsables tengan una idea general de cómo avanza la obtención de las evidencias para estos requerimientos.

34. **Añadir y eliminar evidencias:** Esta funcionalidad permitirá subir uno o varios documentos con las evidencias solicitadas para cada requerimiento de información individual. También permitirá eliminar los documentos subidos, en caso de que se suba algún fichero erróneo los responsables tendrán la opción de deshacer la acción.

35. **Recibir recordatorio:** Esta funcionalidad del sistema enviará un recordatorio de los puntos pendientes a los responsables de IT según la política definida por los auditores. En este recordatorio se adjuntará un enlace a la interfaz del sistema a la que se deben conectar para subir las evidencias solicitadas.

9.- Requisitos del sistema

En este apartado se especificarán tanto los requisitos funcionales como los requisitos no funcionales que deberá cumplir el sistema a desarrollar.

9.1.- Requisitos funcionales - Casos de uso

Los requisitos funcionales que deberá cumplir el sistema se expresarán mediante casos de uso, se ha decidido utilizar esta opción debido a que los casos de uso detallan cómo el usuario interactúa con el sistema y cómo responde este ante esta interacción. De esta forma se define claramente qué debe ofrecer el sistema de una forma comprensible tanto por el equipo que lo diseñará como por los otros stakeholders interesados en éste.

Para facilitar la comprensión de estos casos de uso, se ha decidido relacionarlos con las funcionalidades previamente definidas en el punto *8.2.- Funcionalidades del sistema,* de forma que cada caso de uso responderá a una de las funcionalidades esperadas por los stakeholders.

De cara a visualizar mejor esta relación, se han agrupado los casos de uso en función del actor y de la funcionalidad a la que responde cada caso. A cada uno de los casos de uso se le ha asignado un identificador único y se ha descrito detalladamente en qué consiste.

Los actores del sistema han sido definidos dependiendo de las funcionalidades a las que tendrán acceso. Primero de todo se han identificado dos actores, Responsable IT Cliente y Auditor, correspondientes a los dos tipos de usuario que accederán al sistema. Dentro de los auditores se identifica un perfil estándar, el de *Auditor,* y una especialización de este con más funcionalidades, el perfil *Manager.* Estos perfiles, más los roles de los auditores en una base, que son *Team Leader, Reviewer* o *Team Member* tendrán casos de uso particulares y se tratarán como actores distintos para facilitar la diferenciación entre ellos.

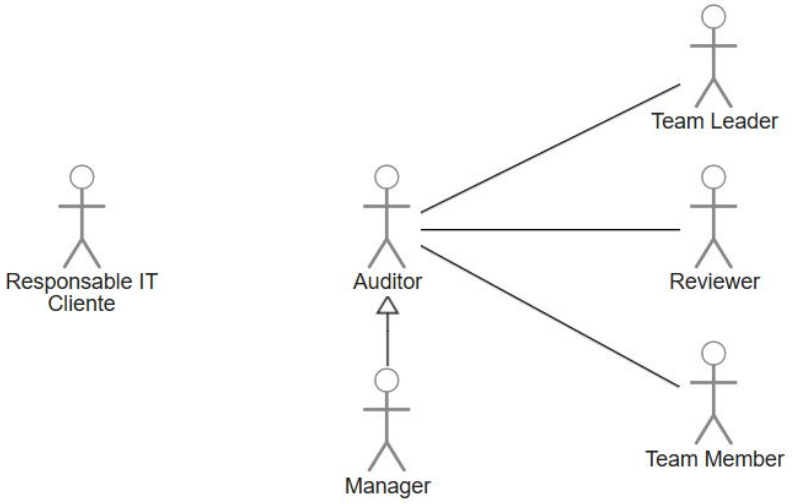


Figura 14. Diagrama con los distintos actores del sistema.

La especificación completa de los casos de uso se encuentra en el punto ***3.-Especificación completa de los casos de uso del sistema*** del **Anexo**. Estos casos de uso se han representado mediante diagramas de casos de uso, los cuales han agrupado los casos representados por funcionalidades relacionadas entre sí. Para cada uno de los casos de uso del diagrama se ha realizado una descripción detallada del comportamiento esperado del sistema.

Se puede observar en el Anexo que se han definido hasta 104 casos de uso distintos que dan respuesta a las funcionalidades definidas y que debe proporcionar el sistema. Esta magnitud de casos de uso permite afirmar que se trata de un sistema bastante complejo y para nada trivial.

9.2.- Requisitos no funcionales

Los requisitos no funcionales, o requisitos de calidad, son aquellos que describen las características que debe cumplir el sistema para realizar las funcionalidades especificadas en los requisitos funcionales, es decir, no describen "qué" debe hacer el sistema, sino "cómo" debe hacerlo.

Los requisitos no funcionales del sistema se han basado en los atributos de calidad definidos en la norma ISO/IEC 25010, que forma parte del estándar ISO/IEC 25000, conocido como SQuaRE (System and Software Quality Requirements and Evaluation), cuyo propósito es el de establecer unos estándares internacionales para la especificación y evaluación de los requerimientos de calidad en los sistemas software[5].

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:

Figura 15. Atributos de calidad definidos por la ISO 25010 [6]

Para cada uno de estos ámbitos se definirán los requisitos no funcionales que apliquen para el sistema a diseñar.

Adecuación Funcional

1.- Completitud funcional

Descripción: El sistema debe de cumplir todas las necesidades y objetivos de los usuarios que lo utilizarán.

Justificación: Es muy importante que todas las necesidades de los auditores se encuentren cubiertas por el sistema, debido a que este se va a desarrollar específicamente debido a que el sistema actual no las cubre.

Condición de satisfacción: Los usuarios finales han validado los requisitos funcionales del sistema y han participado en su especificación, por lo que se considera que las funcionalidades esperadas están correctamente plasmadas. Aún así los usuarios probaran el sistema y analizarán que estén presentes todos estos requerimientos.

Eficiencia

2.- Comportamiento temporal

Descripción: El sistema debe ofrecer unos tiempos de respuesta y de procesamiento aceptables.

Justificación: Uno de los defectos del sistema actual es su lentitud al abrir y ejecutar ciertos procesos, esto es debido a que es un programa antiguo y está poco optimizado. El nuevo sistema está pensado para agilizar y mejorar el trabajo de documentación, por lo que es necesario que los tiempos de respuesta sean cortos y no existan demoras innecesarias.

Condición de satisfacción: Se establecerá un máximo de tiempo de respuesta para cada funcionalidad y luego se medirán los tiempos de ejecución de las funcionalidades durante el desarrollo, una vez desarrollado se volverán a probar estos tiempos en un entorno real.

3.- Capacidad

Descripción: El sistema debe de soportar las conexiones simultáneas de todos los auditores de la firma sin experimentar ralentizaciones o problemas de servicio.

Justificación: Si el sistema no es capaz de ofrecer este requisito no servirá para el dia a dia de la compañía, ya que se necesita que todos los usuarios puedan trabajar a la vez sin problemas.

Condición de satisfacción: Todos los auditores que usarán el sistema se conectarán de forma simultánea y utilizaran sus funcionalidades, si los tiempos de respuesta y las funcionalidades se comportan como es esperado se considerará satisfecho el requisito.

Compatibilidad

4.- Interoperabilidad

Descripción: El sistema tiene que ser capaz de comunicarse con el Servicio de Directorio (SD) de la compañía para autenticar a los usuarios que se conecten en él.

Justificación: Actualmente todos las aplicaciones de la compañía se vinculan con el SD, de forma que los usuarios se identifican en su dispositivo con esas credenciales y a partir de ahí se les garantiza acceso a las otras aplicaciones. No disponer de esta capacidad iría en contra de la política y la forma de funcionar de la compañía.

Condición de satisfacción: El usuario debe poder acceder al sistema identificándose en el SD únicamente, de forma que el sistema se comunique con este para validar su acceso.

Usabilidad

5.- Capacidad para ser usado

Descripción: El sistema tiene que ser fácilmente usable para los usuarios, de forma que se entiendan fácilmente sus funcionalidades y utilizarlas sea relativamente sencillo.

Justificación: Uno de los principales defectos del sistema actual es su sobrecarga de opciones que no tienen ninguna utilidad y la poca intuitividad para realizar ciertas acciones. El nuevo sistema debe corregir estos defectos y ofrecer una buena usabilidad para que los auditores saquen el máximo partido de él y optimicen su tiempo documentando.

Condición de satisfacción: Después de dos semanas de uso del sistema, se realizará una encuesta anónima entre los usuarios para valorar la facilidad de uso una vez familiarizados con el mismo. Si los resultados son positivos se considerará satisfecho el requisito.

6.- Protección contra errores de usuario

Descripción: El sistema debe limitar y controlar las opciones que da a los usuarios para evitar que se puedan cometer errores en el uso del sistema.

Justificación: Uno de los principales defectos del sistema actual es el poco control en la creación y la modificación de elementos, por los que se considera importante que el nuevo sistema controle en detalle la creación y edición de los elementos del sistema de forma que se eviten los errores en la creación y edición de estos.

Condición de satisfacción: El sistema deberá obligar al usuario a seleccionar ciertos elementos de listas predefinidas siempre que sea conveniente (por ejemplo, en la creación de un ITGC concreto seleccionar de la lista con los ITGCs estándares registrados cual es al que representa) y también deberá requerirá una confirmación del usuario cuando se eliminen elementos del sistema para evitar que esta acción se realice por error.

Fiabilidad

7.- Disponibilidad

Descripción: El sistema tiene que estar siempre operativo y accesible durante el horario laboral de la compañía y una gran parte del tiempo fuera de este horario.

Justificación: Si el sistema no está operativo o no puede ser accedido, los auditores no podrán realizar correctamente su trabajo y los clientes no podrán subir la información requerida, por lo que es necesario que este tiempo de baja sea mínimo.

Condición de satisfacción: El sistema se diseñará teniendo en cuenta esta característica y se buscará ofrecer una disponibilidad del 99.9% dentro del horario laboral y un 99% fuera de este. Las actualizaciones y el mantenimiento del sistema se programarán siempre fuera del horario laboral para no entorpecer el trabajo de los auditores.

8.- Capacidad de recuperación

Descripción: El sistema tiene que ser capaz de recuperar toda la información contenida en él en el caso de que se produzca alguna interrupción o fallo.

Justificación: La principal funcionalidad del sistema es la de gestionar la documentación de las auditorías, por lo que la información contenida es el elemento más importante y crítico del sistema, esto implica que en caso de interrupción o fallo del sistema la información contenida no debería verse afectada.

Condición de satisfacción: Se estructurará la base de datos del sistema en un RAID de forma que la información se encuentre duplicada, de esta forma en caso de fallo o interrupción si un disco se viera afectado su información no correría peligro ya que estaría replicada en otro disco.

Seguridad

9.- Confidencialidad

Descripción: El sistema debe proteger los datos y la información que se encuentran almacenados en éste de cualquier acceso no autorizado a ellos.

Justificación: La información con la que se trabaja para auditar los sistemas un cliente es altamente confidencial, el hecho de que una persona ajena a la compañía accediera a esta información supondría un gran problema a nivel legal y para la reputación de la compañía, por lo que es muy importante que el acceso a la información esté controlado.

Condición de satisfacción: Para satisfacer este requisito se diseñará el sistema desde la base teniendo siempre en mente la seguridad, lo que se llama “Secure by design”. El acceso al sistema se realizará siempre dentro de la red privada de la compañía por parte de los auditores y se aplicarán las mejores políticas en cuanto a seguridad en la gestión del sistema.

10.- No repudio

Descripción: El sistema tiene que registrar las acciones o eventos que se produzcan en el sistema, de forma que dichas acciones o eventos no puedan ser repudiados posteriormente.

Justificación: En una base de auditoría trabajan varios auditores a la vez y los EGAs que no están cerrados pueden ser modificados por más de un auditor, aunque estos no tengan el EGA asignado. Idealmente un auditor solo modifica los EGAs que tiene asignados, pero el sistema no limita esta interacción para no lastrar la agilidad del equipo.

Es por esto que se considera importante tener un mecanismo para poder identificar quien ha realizado modificaciones a cada elemento de la base, debido a que, por ejemplo, si alguien accediera a un EGA que no tiene asignado y lo modificara, sin este mecanismo la modificación quedaría desapercibida y no se podría identificar al usuario que la ha realizado y esto podría acarrear muchas consecuencias negativas.

Condición de satisfacción: La base de datos del sistema se diseñará para que se contenga *logs* de auditoría que para cada modificación de un elemento del sistema almacenen qué usuario la ha realizado y en qué fecha y hora.

11.- Autenticidad

Descripción: El sistema tiene que tener registrados a todos los usuarios de forma que pueda identificar la identidad de cada uno de ellos.

Justificación: Todas las acciones que se realicen en el sistema tienen que ser trazables y se tienen que poder identificar a un individuo en concreto, por esto el sistema sólo debe permitir a los usuarios identificados acceder al sistema y tiene que poder trazar sus acciones.

Condición de satisfacción: El sistema no permitirá acceso a usuarios que no estén identificados, además con los *logs* de auditoría identificará debidamente quién es el responsable de cada acción realizada en el sistema.

Mantenibilidad

12.- Modularidad

Descripción: Los componentes que conformen el sistema deben estar diseñados de forma que un cambio interno en un componente tenga un impacto mínimo en los demás

Justificación: Se requiere que el sistema ofrezca una alta cambiabilidad para poder adaptarse a las necesidades que vayan apareciendo dentro del departamento, por lo que la modularidad facilitará mucho realizar cambios en elementos específicos del sistema.

Condición de satisfacción: El sistema se diseñará siguiendo los mejores principios de arquitectura del software para asegurar que se cumple esta modularidad.

13.- Reusabilidad

Descripción: El sistema se tiene que desarrollar de una forma que permita reusar parte de su código para ser utilizado en la construcción de otro software.

Justificación: Aunque el sistema está pensado para la auditoría informática no se puede obviar que la auditoría financiera es el principal negocio de la compañía, por lo que es previsible que se diseñe un nuevo sistema para sustituir a Aura para esta. Teniendo esto en cuenta, es importante diseñar el sistema de forma que se puedan reutilizar sus elementos por si existieran elementos comunes con la auditoría financiera y se quisiera aprovechar el trabajo ya realizado.

Condición de satisfacción: El sistema se diseñará siguiendo los mejores principios de arquitectura del software para asegurar que se las funcionalidades están abstraídas en funciones y clases que permitan su reutilización sin depender del contexto del sistema.

14.- Capacidad para ser modificado

Descripción: El sistema debe de tener una alta cambiabilidad para poder ser fácilmente modificado en el futuro.

Justificación: Se requiere que el sistema ofrezca una alta cambiabilidad para poder adaptarse a las necesidades que vayan apareciendo dentro del departamento.

Condición de satisfacción: El sistema se diseñará siguiendo los mejores principios de arquitectura del software para asegurar que la cambiabilidad del sistema es alta y que añadir nuevas funcionalidades o modificar las existentes no será problemático en el futuro.

Portabilidad

15.- Adaptabilidad

Descripción: El sistema tiene que ser fácilmente adaptable de forma efectiva y eficiente en distintos entornos de hardware y software.

Justificación: Aunque inicialmente el sistema se ejecutará en un servidor en el cloud y los auditores accederán a él a través de su PC en una aplicación web, se requiere que el sistema sea fácilmente adaptable a otros entornos (como dispositivos móviles).Esto es debido a que se quiere dar la opción de desarrollar una aplicación móvil en el futuro para acceder al sistema.

Condición de satisfacción: Se diseñará la arquitectura del sistema de forma que se pueda desarrollar un nuevo tipo de interfaz en el futuro que utilice las mismas funcionalidades que la interfaz actual sin tener que modificar internamente estas funcionalidades.

10.- Modelo conceptual de datos

Para una mejor comprensión del dominio del sistema se ha elaborado un modelo conceptual de datos, el cual plasma todos aquellos conceptos que deberá representar el sistema para cumplir las necesidades de los auditores, las relaciones entre ellos y las restricciones que debe cumplir cada elemento.

10.1.- Diagrama de clases en UML

Este diagrama se ha elaborado teniendo en cuenta las necesidades de los auditores, las funcionalidades que pretende ofrecer el sistema y los elementos ya existentes en el sistema actual. También se han tenido en cuenta los requisitos funcionales y no funcionales del sistema, por lo que todos las clases, relaciones y restricciones se han diseñado para dar respuesta a estos requisitos.

En la siguiente página se encuentra el diagrama de clases diseñado utilizando el lenguaje UML y a continuación se encuentran las restricciones textuales asociadas a este diagrama. Por último se ha añadido una descripción de las clases más relevantes para este dominio particular.

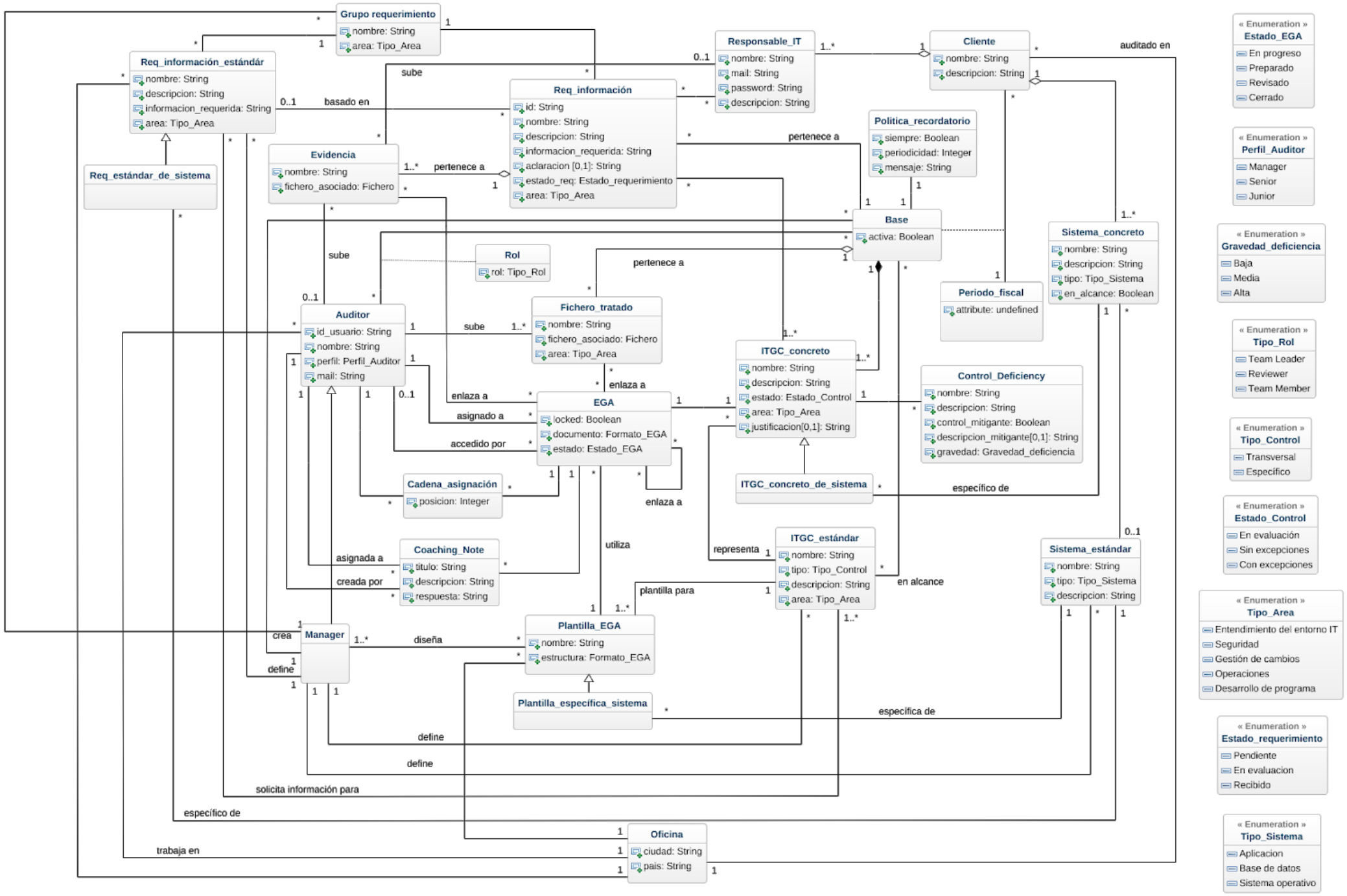


Figura 16. Diagrama de clases de especificación en UML

10.1.1- Restricciones textuales

1. Claves externas:
    - Auditor: id_usuario
    - Responsable_IT: mail
    - Cliente: nombre
    - Base: cliente, periodo_fiscal
    - Oficina: ciudad, país
    - Sistema_estándar: nombre
    - ITGC_estándar: nombre
    - Plantilla_EGA: nombre, ITGC_estándar, oficina
    - Grupo_Requerimiento: nombre, area
    - Req_información_estándar: nombre, Grupo_Requerimiento, oficina
    - Sistema_concreto: nombre, cliente
    - ITGC_concreto: nombre, base
    - Control_Deficiency: ITGC_concreto
    - EGA: ITGC_concreto
    - Cadena_asignación: EGA
    - Coaching_Note: título, EGA
    - Req_información: id, base
    - Evidencia: nombre, req_información
    - Fichero_tratado: nombre, base
    - Politica_recordatorio: base
    - Rol: auditor, base

2. Un cliente solo puede tener asignada una única base para cada período fiscal.

3. El *área* de un *ITGC_concreto* tiene que ser igual a la del *ITGC_estándar* al que representa.

4. Un *ITGC_estándar* que esté en el alcance de una *base* siempre tiene que tener un *ITGC_concreto* que le represente asociado con esta misma *base.*

5. El área y grupo de un *Req_información* son siempre las mismas que las del *Req_información_estándar* en el que se basa, en el caso de éste último exista.

6. Un *ITGC_concreto* sólo puede tener un *Control_Deficiency* asociado si el estado de este ITGC corresponde a *Con excepciones.*

7. Un *Responsable_IT* solo se puede asociar con un *Req_información* si éste corresponde a una *base* asociada al *cliente* del que forma parte el *Responsable_IT* .

8. Un *Responsable_IT* sólo puede subir una *Evidencia* si es para un *Req_información* asociado a él.

9. Un *EGA* solo puede enlazar a otro *EGA* si los *ITGC_concreto* asociados a estos corresponden a la misma *base.*

10. Un *EGA* sólo puede estar relacionado con una *Plantilla_EGA* si su *ITGC_concreto* representa al mismo *ITGC_estándar* con el que está asociada la *Plantilla_EGA.*

11. Un *EGA* sólo se puede relacionar con una *Plantilla_especifica_sistema* si se encuentra asociado con un *ITGC_concreto_de_sistema* cuyo *Sistema_concreto* representa al mismo *Sistema_estándar* con el que se relaciona la *Plantilla_especifica_sistema.*

12. Un *EGA* sólo puede estar relacionado con una *Plantilla_EGA* si el *cliente* de la *base* correspondiente a su *ITGC_concreto* es auditado por la misma *oficina* con la que está relacionada la *Plantilla_EGA.*

13. Un *EGA* sólo se puede relacionar con un *Fichero_tratado* si su *ITGC_concreto* asociado se corresponde a la misma *base* de la que pertenece el *Fichero_tratado.*

14. Un *EGA* sólo se puede relacionar con una *Evidencia* si el *ITGC_concreto* de este *EGA* se corresponde a la misma *base* de la que se pertenece el *Req_información* que contiene la *Evidencia.*

15. Un *Auditor* sólo se puede relacionar con un *EGA* si el *ITGC_concreto* de este *EGA* corresponde a una *base* asignada al *Auditor.*

16. Un *Auditor* sólo se puede relacionar con una *Cadena_asignación* si el *ITGC_concreto* del *EGA* asociado a esta se corresponde a una *base* asignada al *Auditor*.

17. Un *Auditor* sólo se puede relacionar con una *Coaching_Note* si el *ITGC_concreto* del *EGA* asociado a la *Coaching_Note* corresponde a una *base* asignada al *Auditor.*

18. Una *Evidencia* sólo puede ser subida por un *Auditor* si este se encuentra asignado a la misma *base* de la que pertenece el *Req_información* que contiene la *Evidencia.*

19. Un *Auditor* solo *Fichero_tratado* sólo puede ser subido por un *Auditor* si este se encuentra asignado a la misma *base* de la que se pertenece el *Fichero_tratado*.

20. Un *Req_información* basado en un *Req_estándar_de_sistema* sólo se puede relacionar con un *ITGC_concreto_de_sistema* cuyo *Sistema_concreto* represente al mismo *Sistema_estándar* asociado con el *Req_estándar_de_sistema.*

21. Un *Req_información* de una *base* concreta sólo se puede basar en un *Req_información_estándar* que esté asociado con la *oficina* en la que se audita al *cliente* de esa misma *base.*

22. Un *Manager* sólo puede crear una *base* para un *cliente* que se audite en la misma *oficina* en la que trabaja.

23. Un *Auditor* sólo puede pertenecer a la subclase *Manager* si su perfil es de tipo “Gerente”.

24. La *oficina* con la que se asocia una *Plantilla_EGA* siempre es la *oficina* en la que trabaja el *Manager* que lo define.

25. La *oficina* con la que se asocia un *Req_información_estándar* siempre es la *oficina* en la que trabaja el *Manager* que lo define.

26. Un *Req_información* sólo se puede asociar con un *ITGC_concreto* que se corresponda a la misma *base* de la que pertenece el *Req_información.*

27. Una *Plantilla_especifica_sistema* sólo puede estar asociada con un *ITGC_estándar* de tipo *Específico.*

28. Un *ITGC_concreto_de_sistema* sólo puede estar relacionado con un *ITGC_estándar* de tipo *De sistema.*

29. Sólo podrá existir un único *ITGC_concreto_de_sistema* por cada combinación entre su *ITGC_estándar* y el *Sistema_concreto* con el que se relaciona.

10.2.- Detalles de las clases del diagrama

En este apartado se concretarán los detalles de las clases más relevantes del sistema y también se explicará qué conceptos representan.

Auditor

Representa a los auditores de sistemas de la firma que son los encargados de realizar las propias auditorías. Estos auditores tienen un perfil asignado en función de su rango en el departamento, los perfiles existentes son los siguientes: Auditor, Senior y Gerente. Existe la subclase *Manager* que está formada por aquellos auditores con rango “Gerente” y que tienen la responsabilidad de gestionar las bases del sistema.

Base

Representa una auditoría realizada para un cliente auditado durante un año fiscal concreto. Esta auditoría la realiza el equipo de auditores que trabajan en la oficina de la firma que tiene asignado ese cliente. Todos los elementos específicos de esa auditoría concreta están relacionados con esta base y sólo pueden ser accedidos por los auditores a los que se les ha garantizado acceso a la misma. La base puede encontrarse en dos estados: *Activa* o *Cerrada*. El estado *Activa* permite que los auditores de la base modifiquen los elementos específicos de la base (EGAs, ITGCs…) y el estado *Cerrada* sólo permite la consulta de estos elementos, pero no admite ninguna modificación de estos.

Sistema estándar

Representa los sistema estándares genéricos, como podrían ser el ERP SAP o una base de datos Oracle. Esto implica que no representa la implementación concreta que apliquen los clientes para estos sistemas, sino que sirve como registro de aquellos sistemas que se consideran estándares por el alto número de clientes que los implementan (y personalizan) en sus organizaciones. Esto nos permitirá definir plantillas personalizadas y requerimientos de información específicos para estos sistemas.

ITGC estándar

Representa los Controles Generales Informáticos definidos por la ISACA y adaptados por la firma para sus auditorías de sistemas. Esta clase representa el concepto de estos ITGCs a testear en las auditorías, pero no representa la aplicación de ese ITGC en las auditorías de los clientes, de forma que en esta clase se representa el ITGC estandarizado a testear en los clientes y en la clase de *ITGC_Concreto* se representa la aplicación de uno de estos ITGCs en la auditoría de un cliente concreto.

Cada uno de estos ITGCs tiene un nombre específico y contiene una descripción detallada en la que se explicará en qué consiste el ITGC, también tiene asignada una área de aplicación concreta y puede ser de dos tipos distintos. Los de tipo *De sistema* son aquellos que se aplican a nivel de sistema, por lo que se permite testear el mismo ITGC para varios sistemas distintos en una misma auditoría utilizando plantillas distintas para cada sistema. Los ITGCs de tipo *Transversal* se aplican a toda la organización e incluyen a todos los sistemas del cliente por igual, por lo que sólo se testean una vez por auditoría.

Sistema concreto

Representa los sistema concretos que son utilizados por los clientes y que se encuentran en alcance de la auditoría informática. Cada uno de estos sistemas puede ser o bien una implementación de alguno de los sistema estándares registrados, o bien un sistema específico del cliente que no se relacione con ninguno de estos sistemas registrados.

ITGC concreto

Representa la aplicación de los ITGCs estándares a las distintas auditorías que se realizan en el departamento. Cada uno de estos ITGCs concretos está asignado a una única base, que representa la auditoría para la que se aplica el ITGC, también está relacionado directamente con un único ITGC estándar, que corresponde al Control General Informático que se pretende testear con este ITGC concreto en esa auditoría concreta. El testeo de un ITGC concreto será siempre documentado en un único EGA con el que se relacionarà.

Dependiendo del tipo de ITGC estándar, los ITGCs concretos que lo apliquen tendrán características distintas.

- En el caso de que el ITGC estándar sea de tipo *Transversal*, sólo podrá existir un ITGC concreto en la base representando a ese ITGC estándar.
- En el caso de que el ITGC estándar sea de tipo *De sistema*, podrán existir múltiples ITGCs concretos de la subclase *ITGC Concreto de sistema* que se asocien con los múltiples sistemas concretos para los que se quiera testear el control. Adicionalmente también puede existir un ITGC concreto que no esté asignado a ningún sistema y que cubra la parte transversal de ese ITGC.

EGA

Representa los documentos en los que se documenta el trabajo realizado por los auditores en relación con el testeo de la efectividad de un ITGC concreto para un cliente concreto. Cada EGA se asocia a un único ITGC concreto y es documentado y revisado por varios auditores que se asignan en su cadena de revisión. La estructura del EGA es el documento en sí, en el cual se redactan los procedimientos seguidos por el equipo de auditoría y las conclusiones extraídas al aplicar este procedimiento, en este documento también se pueden añadir links o referencias a otros EGAs o a los documentos a partir de los que se han extraído las conclusiones (Evidencias y Ficheros Tratados). Actualmente, esta estructura corresponde a un fichero de tipo *xlsx* que es editado utilizando el programa *Microsoft Excel*.

La estructura de un EGA se añadirá como un como un atributo de la clase correspondiente al fichero que la almacena, pero se deja a valorar por el equipo encargado de su desarrollo de sí seguir utilizando el mismo formato o desarrollar un formato propio (por ejemplo, un formato basado en XML o HTML) interno del sistema que permita representar a un EGA.

En el EGA también se registra su estado, que puede ser *En progreso, Preparado, Revisado* o *Cerrado,* en función del trabajo realizado por los auditores. El EGA también registrará si se encuentra bloqueado o no, ya que cuando un auditor se encuentre editando un EGA este se bloqueará para que no pueda ser editado de forma simultánea, evitando así la generación de conflictos.

Plantilla EGA

Representa las plantillas a partir de las cuales se documentan los EGAs asociados a los ITGCs de una base. Estas se encuentran asociadas siempre a un ITGC estándar y sirven como plantilla para los EGAs que documenten la aplicación de ese ITGC en una base concreta. Existen también plantillas específicas para la aplicación de ese ITGC estándar en un sistema concreto, para que eso sea posible el ITGC estándar tiene que ser de tipo *De sistema*.

La plantilla consta de un nombre que la identifica y de una estructura que estará en el mismo formato en el que se documenten los EGAs. La plantilla será diseñada por los *Managers* del departamento, proveyendo así de un marco con los elementos y procedimientos que se deben seguir para el correcto testeo del ITGC estándar seleccionado, en esta estructura se podrán añadir títulos, texto, tablas… De esta forma, al crear un EGA se seleccionará la plantilla correspondiente a ese ITGC de la que partir y se copiará la su estructura en el EGA, consiguiendo así que el auditor solo tenga que completar la estructura con la información y las conclusiones que obtenga de la realización del trabajo de auditoría y no deba perder el tiempo en diseñar la estructura de ese documento. Además el hecho de que la plantilla establece claramente los elementos y procedimientos a testear en cada control es algo de gran ayuda para los auditores más juniors.

Requerimiento de información estándar

Representa los requerimientos más comunes que realizan los auditores de sistemas a las compañías auditadas en los que se especifica la información necesaria para la correcta realización de las auditorías. Estos requerimientos se definen para servir como marco a aplicar a cada auditoría concreta y sirven como modelo de los requerimientos de información concretos que son los que serán recibidos por los clientes. Estos requerimientos guardan una estrecha relación con los ITGCs genéricos, debido a que la información que se solicitan los auditores siempre se utiliza para testear la efectividad de alguno de los ITGCs genéricos definidos por la firma, es por eso que el área de aplicación del requerimiento debe de coincidir con la de los ITGCs estándares.

Existen también requerimientos que son específicos de un sistema estándar, debido a que cada sistema tiene sus particularidades, por lo que existe una subclase para los requerimientos estándares de sistema, los cuales se relacionan con el sistema estándar para el que solicitan información.

Estos requerimientos están formados por un nombre que los identifica, una descripción que explica en qué consiste el requerimiento, la área de aplicación en la que se engloba el requerimiento y la descripción detallada de la información que se espera recibir para ese requerimiento. Una vez son creados, pueden ser asignados a los responsables de IT del cliente que estén encargados de proporcionar las evidencias relacionadas con ese requerimiento.

Requerimiento de información

Representa un requerimiento de cierta información realizado a un cliente concreto que se encuentra asociado a uno o varios ITGCs de la base de ese cliente. Puede corresponderse, o no, a un requerimiento de información estándar. El sistema ofrecerá el listado de requerimientos estándares que se pueden importar a la base actual en función de los ITGCs concretos presentes en la base, ya que los ITGCs de la base con los que se encuentre asociados deberán ser los mismos que los ITGCs estándares con los que se asocia el requerimiento estándar. Al crear el requerimiento basándose en un requerimiento estándar, su nombre, descripción, información requerida y área de aplicación se inicializarán con los mismos valores de este. El identificador del requerimiento será un campo que contendrá un texto compuesto por dígitos separados por puntos (“1.1” o “2.3.1” por ejemplo), donde el primer dígito identificará siempre el área de aplicación del requerimiento y los siguientes a los requerimientos en sí, permitiendo asignar .

En el caso de en la base se encuentren presentes ITGCs de sistemas estándares y que estos ITGCs tengan requerimientos estándares de sistema asociados, estos requerimientos estándares también aparecerán como importables.

El usuario podrá modificar manualmente estos campos, y podrá añadir el campo aclaración en el cual se resolverán dudas específicas que tengan los clientes para ese requerimiento concreto. Los requerimientos también tendrán un estado, que se irá actualizando a medida que los clientes proporcionen las evidencias de información solicitadas en estos requerimientos.

Evidencia

Representa un documento que contiene la información, o parte de ella, solicitada por los auditores en un requerimiento de información sobre las políticas o los sistemas de un cliente concreto para una auditoría concreta. Esta evidencia puede ser subida directamente por los responsables de IT del cliente, vía la interfaz del sistema a la que tienen acceso, o por los propios auditores, en el caso de que reciban la información a través de otros medios o la extraigan directamente con los medios presentes en la oficina del cliente.

Están compuestas por un nombre que las identifica, el fichero que contiene la evidencia (por ejemplo un documento de texto, de hojas de cálculo o incluso una captura de pantalla) y se asocia siempre a un único requerimiento de información. Estas evidencias luego son linkeadas en los EGAs que utilicen la información que contienen para fundamentar las conclusiones obtenidas sobre los ITGCs evaluados.

Fichero tratado

Representa un documento que ha sido tratado por los auditores, cuya información inicial estaba contenida en una o varias evidencias, para poder analizar mejor la información contenida en estas. Estos ficheros son subidos únicamente por los auditores, que son quienes los crean y modifican. Contienen un nombre que los identifica y se les asigna una área de aplicación para organizarlos dentro de la base. Al igual que las evidencias, estos luego son enlazados en los EGAs que utilizan la información que contienen para fundamentar las conclusiones obtenidas sobre los ITGCs evaluados.

Rol

Representa el rol con el que un auditor es asignado a una base concreta. Este rol puede ser el de *Team Member, Reviewer* o *Team leader,* dependiendo del rol asignado, las opciones y funcionalidades a las que tendrá acceso el auditor dentro de la base serán distintas.

Cadena asignación

Representa la relación entre un EGA concreto y aquellos auditores que se encargarán de completarlo y revisarlo. La cadena contiene el orden en el que se encuentra asignado el EGA a estos auditores, de forma que el primer auditor en la cadena será el encargado de completarlo y los siguientes auditores se encargarán de revisar el trabajo realizado, en cuanto uno marque como revisado el EGA, este pasará al siguiente auditor hasta llegar al último de la cadena.

Coaching note

Representa una nota que se puede crear para apuntar información relacionada con un EGA concreto. Esta nota puede ser creada por cualquier auditor que esté en la cadena de asignación del EGA y en el momento de la creación se tiene que elegir a otro auditor de la cadena (que puede ser uno mismo) al que se le asignará la nota. Este concepto es útil para apuntar cosas a tener en cuenta de cara a preparar un EGA, asignándose la nota a uno mismo en este caso; o para que un revisor de feedback sobre el trabajo realizado al auditor que ha preparado el EGA, asignándole la nota a ese auditor en este caso.

Control deficiency

Representa una deficiencia encontrada en la aplicación de un ITGC concreto en la auditoría de un cliente. Este concepto se crea para tener registradas estas deficiencias en un sitio aparte de los EGAs, donde son analizadas y justificadas basándose en las evidencias proporcionadas. De esta forma, el auditor encargado de evaluar el estado de los sistemas del cliente y de emitir el documento con las recomendaciones a seguir puede acceder directamente a estas deficiencias sin necesidad de abrir todos los EGAs y analizar su contenido.

Una deficiencia consta de una descripción, en la que se detalla en qué consiste, un campo que nos indica si existe o no un control mitigante (que no es nada más que otro control realizado que permite mitigar el riesgo de que esta deficiencia pueda tener impacto en los estados financieros de la compañía), la descripción del control mitigante realizado en caso de que exista y, por último, se clasifica esta deficiencia en función de su gravedad.

Parte 2

-

Fase de diseño

11.- Diseño arquitectónico

En este apartado se tratará el diseño arquitectónico del sistema desde un punto de vista general, analizando la arquitectura propuesta, tanto la parte física como lógica, y recomendando las tecnologías y los patrones más adecuados para satisfacer las necesidades de los usuarios y cumplir con los requisitos especificados, pero sin llegar a entrar en detalles del diseño funcional del sistema necesario para implementar los casos de uso especificados en la fase anterior.

Se ha decidido optar por esta vía debido a que la implementación del sistema no se encuentra en el alcance del proyecto, sino que esta será realizada por el socio tecnológico de la compañía, por lo que se ha considerado pertinente dar la libertad a este actor de realizar el diseño funcional que considere más conveniente. Si lo considera conveniente, podrá partir de la arquitectura recomendada en este apartado y extenderla para implementar el sistema, por lo que esta arquitectura debe de interpretarse como una propuesta de partida que el desarrollador deberá concretar para realizar la implementación.

11.1.- Visión general de la arquitectura

El diseño propuesto para el sistema es una aplicación distribuida, de forma que parte del código se ejecutará en un servidor de *backend* y el otro en dos aplicaciones web de *frontend,* una para los auditores y otra para los responsables de IT de los clientes.

El servidor de *backend* del sistema se encargará de gestionar e implementar toda la lógica de negocio del sistema y de almacenar la información en una base de datos. Responderá y procesará los *requests* que le realicen las aplicaciones de *frontend*, pero los usuarios finales no tendrán acceso directo a este elemento en ningún momento.

Las dos aplicaciones web de *frontend* se encargarán de gestionar las interfaces de usuario que utilizarán los auditores y los responsables de IT de los clientes respectivamente para acceder al sistema. Estas aplicaciones deberán mostrar los elementos del sistema a los usuarios, gestionar su interacción con ellos y capturar los datos de entrada que introduzcan estos. No tendrán conocimientos sobre la lógica de negocio del sistema ni de los datos almacenados por este, por lo que se comunicarán con el servidor de *backend* para procesar la información introducida por el usuario y para solicitar la información que debe mostrar en cada vista.

Estas dos aplicaciones web para el *frontend* serán de tipo *Single-page application (SPA),* lo que significa que se cargarán en el navegador una sola vez e irán reescribiendo el código de la vista de forma dinámica en función de las interacciones con el usuario. También solicitarán de forma dinámica los recursos necesarios para mostrar en cada vista al servidor de *backend.*

Se ha decidido separar el sistema en las partes de *frontend* y *backend* para así desacoplar la interfaz de usuario del sistema de la lógica de negocios y el modelo de datos, de esta forma el diseño e implementación de ambos elementos es más eficiente al no depender tanto el uno del otro.

Esta eficiencia es debido a que para conectar el servidor de *backend* con las aplicaciones de *frontend* se utiliza una interfaz (vía APIs REST o similares) en la que se define cómo se comunican ambos elementos y en esta interfaz se definen los 'contratos' de las peticiones que accepta el servidor de *backend,* especificando para cada petición los datos que espera recibir y los resultados que devolverá al procesarse. De esta forma, los desarrolladores del *backend* tienen libertad en su desarrollo mientras puedan satisfacer los contratos de las peticiones, y los desarrolladores del *frontend* pueden diseñar las interfaces como ellos consideren adecuado teniendo en cuenta que deberán utilizar dichas peticiones para comunicarse con el backend.

11.2.- Arquitectura lógica

La arquitectura del sistema se va a basar en el patrón arquitectónico MVC (Modelo - Vista - Controlador). Este patrón es muy utilizado en el desarrollo de aplicaciones web y existen una gran variedad de *frameworks* que lo implementan. Este patrón, tal y como indica su nombre, divide el sistema en tres componentes distintos:

- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones e implementando también la lógica de negocio. Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

- **Vista:** Presenta el 'modelo' en un formato adecuado para interactuar, usualmente la interfaz de usuario, por tanto necesita obtener de dicho 'modelo' la información que debe representar como salida.

- **Controlador:** Responde a eventos e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

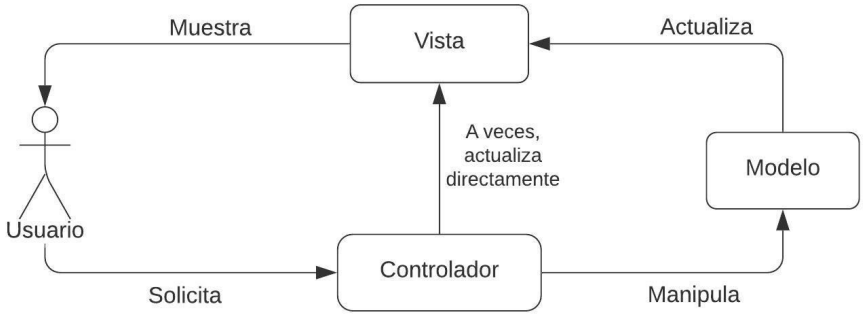


Figura 17. Esquema del comportamiento típico del patrón MVC

En el caso del sistema propuesto, la aplicación de este patrón no es trivial debido a que la aplicación se encuentra distribuida entre un servidor de *backend* y dos aplicaciones web de *frontend.* Uno podría pensar que el componente de la 'vista' y del 'controlador' se encuentran en las aplicaciones de *frontend* y que el componente del 'modelo' se encuentra en el servidor de *backend,* pero esto implicaría que el 'controlador' de las aplicaciones de *frontend* invocaría directamente las peticiones sobre el 'modelo' del servidor y éste actualizaría directamente a la 'vista' del *frontend*, generando así un gran acoplamiento entre ambos, que es justamente lo que se pretende evitar al utilizar esta arquitectura.

Para solucionar este problema, se aplicará el patrón MVC por separado tanto en las aplicaciones de *frontend* como para el *servidor* de backend, de forma que los dos elementos tendrán su propio 'modelo', 'controlador' y 'vista'. Las dos aplicaciones web seguirán la misma arquitectura, sólo se diferenciarán en las funcionalidades que ofrecerán.

En las aplicaciones web de *frontend* los componentes serán los siguientes:

- El 'modelo' contendrá la representación de la información a mostrar y la lógica de presentación. Recibirá las solicitudes del 'controlador' y a su vez solicitará los recursos que necesite en cada momento al servidor de *backend* mediante una interfaz entre ambos, estos se utilizarán para representar la información del sistema y actualizar la 'vista' de forma dinámica. La representación de la información será un subconjunto normalizado del 'modelo' presente en el servidor de *backend*.
  Este componente es necesario debido que al ser una aplicación web de tipo *SPA* las vistas son generadas de forma dinámica, por lo que es necesario disponer de una lógica de presentación que defina cómo se generan y que solicite los recursos necesarias para generarlas.

- La 'vista' se encargará de construir y mostrar la interfaz del sistema al usuario. Será actualizada de forma dinámica por parte del 'modelo'.

- El 'controlador' registrará las interacciones del usuario y los parámetros de entrada que este introduzca. En función de los elementos seleccionados y/o introducidos por el usuario esté invocará la petición correspondiente en el 'modelo'.

En el servidor de *backend* los componentes tendrán las siguientes funcionalidades:

- El 'modelo' contendrá la lógica de negocio y la representación del dominio del sistema. Recibirá las peticiones del 'controlador' para actualizar este modelo con la información recibida desde la aplicación web y les dará respuesta aplicando la lógica de negocio. También recibirá *querys* de la 'vista' para obtener cierta información representada en este modelo y les dará respuesta aplicando la lógica de negocio.

- La 'vista' generará la respuesta a las peticiones del 'modelo' de las aplicaciones web en un formato que este pueda procesar, pero sin generar en ningún momento el código *HTML* a mostrar por estas. En caso de que en esta respuesta necesite proporcionar elementos almacenados en el sistema (por ejemplo, obtener todos los ITGCs estándares registrados en el sistema), la 'vista' realizará una petición al 'modelo' para obtenerlos y con ellos construirá la respuesta para la aplicación web.

- El 'controlador' recibirá las peticiones del 'modelo' de las aplicaciones web. En función de la petición, opcionalmente actualizará el 'modelo' con los parámetros enviados junto a esta y luego cederá el control a la 'vista' para que genere la respuesta a la petición.

Esta arquitectura se encuentra representada gráficamente en el siguiente diagrama, donde el *Client* representa las aplicaciones web de *frontend* y el *Server* al servidor de *backend*:

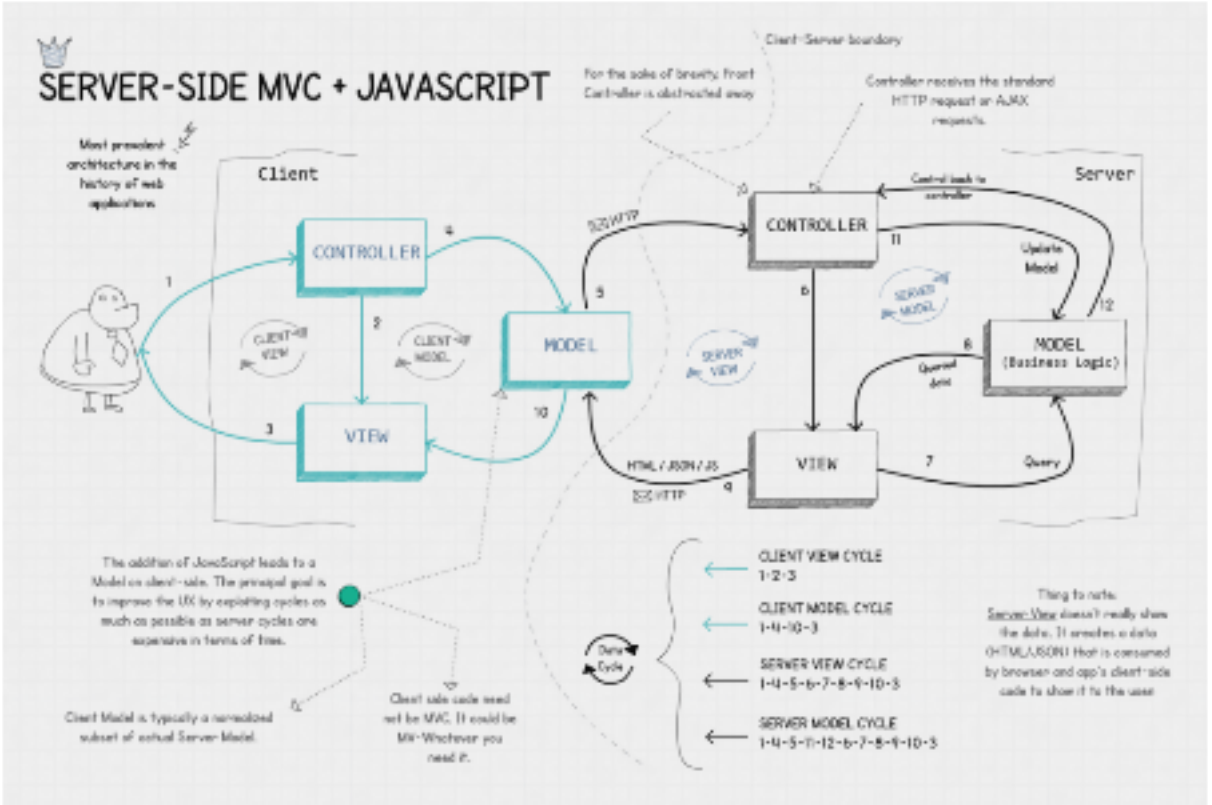


Figura 18. Representación gráfica de la aplicación del patrón MVC en una arquitectura cliente servidor con una aplicación web de tipo SPA desarrollada en Javascript [7]

En el diagrama aparece representado cómo interaccionan entre ellos los distintos componentes del sistema para procesar y satisfacer las solicitudes de los usuarios. Pero para entender mejor el flujo que sigue el sistema, desde la petición inicial del usuario hasta que el sistema le muestra su respuesta, se va a utilizar un ejemplo concreto del dominio del sistema a desarrollar.

Antes de analizar este flujo, se tiene que tener presente que la comunicación entre cliente y servidor se realiza de forma asíncrona, por lo que el cliente puede hacer una request al servidor y seguir ejecutando parte del código mientra espera su respuesta. Para simplificar este hecho y facilitar la comprensión, se asumirá que los pasos que seguirá el sistema entre sus componentes serán secuenciales, pero se tiene que tener en cuenta que en el mundo real algunos de estos pasos se podrían empezar a ejecutar antes de que finalizaran sus pasos previos.

También se tiene que tener en cuenta que los pasos se identificarán con el mismo número con el que aparecen en el diagrama y se ordenarán de forma cronológica.

Ahora supongamos que el usuario se encuentra dentro de una base concreta y decide abrir la vista correspondiente al “Dashboard”, donde se muestran todos los EGAs que tiene ese usuario asignado.

El flujo que seguiría el sistema entre los componentes es el siguiente:

- **Usuario → Controlador cliente (1):** El usuario selecciona la opción *Dashboard* en la interfaz y el controlador del cliente registra esta interacción.

- **Controlador cliente → Modelo cliente (4):** El controlador del cliente registra que el usuario quiere abrir el *Dashboard* en cierta base, al tratarse de una opción cuya respuesta es una nueva vista con nuevos elementos del sistema, solicita al modelo del cliente que resuelva la petición del usuario y le envía el identificador de la base y también le indica qué usuario realiza la petición.

- **Modelo cliente → Controlador servidor (5):** El modelo del cliente recibe la solicitud para obtener los recursos necesarios a mostrar en la vista del *Dashboard* de cierta base para un usuario determinado. Debido a que esta información se encuentra en el modelo del servidor, se realiza una petición al controlador del servidor mediante la interfaz entre ambos en la que se le indica la base y el usuario para el que se quiere abrir el *Dashboard* y se le solicitan los recursos necesarios para abrir esa vista.

- **Controlador servidor → Vista servidor (6):** El controlador del servidor recibe la solicitud y al no tratarse de ninguna modificación del modelo actual, sino una consulta, no solicita al modelo ninguna actualización sino que solicita directamente a la vista del servidor que genere la respuesta a esa petición.

- **Vista servidor → Modelo servidor (7):** La vista del servidor recibe la solicitud para generar la respuesta a la petición recibida, que consiste en todos los EGAs, y su información asociada, asignados al usuario que los ha solicitado. Al no disponer la vista de esta información, hace una *Query* al modelo del servidor en la que le solicita los recursos necesarios para generar la respuesta para la opción del *Dashboard* de la base y para el usuario recibidos en la petición.

- **Modelo servidor → Vista servidor (8):** El modelo del servidor recibe la *query* y este la procesa. Primero comprueba que el usuario que solicita la vista del *Dashboard* esté asignado a la base para la que lo solicita, una vez comprobado, el modelo accede a la base de datos y obtiene todos los EGAs, con sus CNs asociadas y su cadena de asignación, que correspondan a la base del sistema y estén asignados a ese usuario. Una vez obtenidos le devuelve esta información a la vista del servidor.

- **Vista servidor → Modelo cliente (9):** La vista del servidor recibe el resultado de la query, pero para generar la respuesta, primero debe de formatear dicho resultado al formato que espera recibir el cliente (por ejemplo, al formato JSON). Una vez formateado, la vista responde a la petición inicial que realizó el modelo del cliente con dichos resultados formateados en el cuerpo de la respuesta.

- **Modelo cliente → Vista cliente (10):** El modelo del cliente recibe en la respuesta de la petición los EGAs, con sus CNs asociadas y su cadena de asignación, necesarios para generar la vista del dashboard en el formato acordado con el servidor. El modelo entonces procesa dicha respuesta y genera su propia representación de esa información en su formato interno. Una vez finalizado, el modelo solicita a la vista del cliente que genere la vista del *Dashboard* con esa información concreta.

- **Vista cliente → Usuario (3):** La vista del cliente obtiene de su modelo los datos necesarios para actualizar la interfaz de usuario para que muestre por pantalla los EGAs y su información asociada del usuario en esa base. Con estos datos genera de forma dinámica el código que ejecutará el navegador del usuario y que le será mostrado.

Este flujo entre componente detallado corresponde a la funcionalidad concreta de consultar el *Dashboard,* es interesante destacar que en este flujo no se encuentran representadas todas las interacciones posibles entre componentes, pues se puede observar que las interacciones que aparecen con el número 2, 11 y 12 respectivamente en la *Figura 18* no se encuentran en el flujo de esta funcionalidad.

Respecto al paso número 2, del controlador del cliente a la vista del cliente, esto es debido a que esta sólo tiene lugar si la interacción del usuario registrada por el controlador no requiere de acceso al modelo al ser una petición trivial para modificar la vista actual. Un ejemplo podría ser mostrar un menú cualquiera de la base o desplegar los ITGCs agrupados en una área de aplicación.

Respecto a los pasos 11 y 12, entre el controlador del servidor y el modelo del servidor, esto es debido a que estos pasos sólo se ejecutan si la petición realizada por el cliente implica una modificación del modelo actual. Por ejemplo la creación de un EGA o la modificación del nombre de una Evidencia o Fichero tratado. En estos pasos el controlador solicitaría la actualización al modelo, este procesaría la solicitud aplicando la lógica de negocio correspondiente y una vez finalizado le respondería al controlador confirmando su ejecución o informando del error encontrado al tratar de actualizar el modelo.

11.3.- Arquitectura física

En este apartado se tratará la arquitectura física sobre la que se implementará la arquitectura lógica definida en el apartado anterior. Esta arquitectura física consiste en el hardware sobre el que se ejecutarán las aplicaciones que conforman el sistema y la distribución de este hardware en las redes en las que estarán conectados. Se deberá tratar también cómo se conectarán los usuarios a estos dispositivos y cómo se comunicarán las aplicaciones web de *frontend* con el servidor de *backend.*

11.3.1.- Descripción de la arquitectura

Esta arquitectura se ha diseñado para que tanto las aplicaciones web como el servidor de *backend* se desplieguen y ejecuten en máquinas virtuales en el *cloud* y que los usuarios accedan al sistema mediante un navegador web compatible. De esta forma los usuarios no necesitarán de ningún software específico que se ejecute en sus dispositivos, debido a que todas las funcionalidades se ejecutarán en la nube, sino que simplemente deberán conectarse al sistema a través de la red.

Las aplicaciones web de *frontend* y el servidor de *backend* se conectarán entre ellas mediante una interfaz basada en APIs REST bajo el protocolo HTTPS. De esta forma, las aplicaciones de *frontend* utilizarán esta interfaz para realizar las peticiones que necesiten al servidor de *backend*, accedIendo así a sus recursos y funcionalidades. Esto permite que el servidor y las aplicaciones se encuentren muy desacoplados, de forma que el servidor ofrece un listado definido de peticiones que se le pueden realizar y luego cada aplicación de *frontend* utiliza las que considere convenientes para ofrecer sus funcionalidades, pudiéndose utilizar sin ningún problema la misma petición en distintas aplicaciones.

Para acceder al sistema los usuarios se conectarán a la aplicación web de *frontend,* desplegada en el *cloud,* que les corresponda. De forma que el grupo de usuarios correspondiente a los auditores de y el grupo de los responsables de IT dispondrán cada uno de una aplicación web distinta para acceder al sistema. Estas dos aplicaciones se encontrarán hospedadas en máquinas virtuales distintas y se diferenciarán en las funcionalidades que ofrecerán, la forma de identificar a los usuarios y las redes mediante las que se accederá a ellas, adaptándose así a las distintas necesidades de estos grupos.

Aplicación web - Auditores

La aplicación web destinada a los auditores sólo será accesible dentro de la red interna de la compañía, lo que permitirá tener un mejor control del tráfico con el sistema. Para implementar esta restricción, existirá un servidor *proxy* dentro de esta red que actuará como intermediario entre los dispositivos de los auditores y la aplicación web del sistema, de forma que esta aplicación sólo aceptará las conexiones provenientes de este servidor, asegurando así que todas las conexiones se realizan desde dentro de la red y pueden ser monitoreadas por el equipo de IT de la firma. Esto quiere decir que los auditores deberán estar conectados a la red disponible en las oficinas o en su lugar utilizar una VPN para conectarse a esta red en el caso de que se encuentren fuera de estas.
La aplicación de los auditores también se deberá conectar al servidor de directorio (SD) de la compañía, encontrado dentro de esta red privada, para identificar a los auditores que intenten acceder a la aplicación. Los auditores enviarán sus credenciales a la aplicación web y esta las validará contra el SD de la compañía, en caso de que las credenciales sean correctas el auditor será identificado en el sistema. Todas estas conexiones tendrán lugar bajo el protocolo HTTPS, que asegura que la información se encuentra encriptada y protegida ante cualquier atacante que intente interceptarla.

Aplicación web - Responsables IT Clientes

La aplicación web que utilizarán los responsables de IT de los clientes, a diferencia de la de los auditores, si que podrá ser accedida desde cualquier dispositivo independientemente de la red en la que se encuentre. Al ser una aplicación pensada únicamente para el personal de los clientes, no es para nada interesante que esta aplicación se encuentre visible en internet para el público en general, ya que esto podría atraer la vista de ciberdelincuentes que quisieran atacar al sistema y intentar acceder a la información de los clientes. Para evitar este escenario esta aplicación web no se registrará con ningún dominio, sino que para acceder a ella los usuarios recibirán en sus correos un link con la dirección IP de la aplicación y con este se les abrirá la página inicial para identificarse en el sistema. De esta forma, los posibles atacantes no podrán saber dónde se encuentra esta aplicación hospedada, a no ser que obtengan su dirección IP a partir de los correos de algún cliente, reduciendo así en gran medida el riesgo a sufrir un ataque, ya sea para robar o encriptar la información o de denegación de servicio (DDoS)

Para identificar a los usuarios, estos tendrán que introducir su dirección de email corporativo y luego introducir su contraseña. Estas conexiones tendrán lugar bajo el protocolo HTTPS, garantizando así que la información que suben los usuarios al sistema se encuentra encriptada y protegida ante cualquier atacante que intente interceptarla.

Servidor *backend*

El servidor de *backend* se encontrará desplegado en el *cloud* igual que las aplicaciones web de *frontend*, pero a diferencia de estas, este servidor no será accedido directamente por los usuarios, los cuales desconocerán completamente su ubicación. Sólo las aplicaciones de *frontend* comentadas anteriormente conocerán la dirección del servidor y podrán acceder a este mediante una interfaz basada en APIs REST.

El servidor de *backend* será el único elemento que dispondrá de acceso a la base de datos del sistema, que también se encontrará desplegada en el *cloud*, y que almacenará toda la información del sistema (bases, EGAs, evidencias, ITGCs, etc.) El servidor consultará y modificará la base de datos en función de la lógica de negocio que tenga implementada.

Esta base de datos se encontrará en el *cloud* del mismo proveedor que el servidor de *backend* por motivos de practicidad, ya que estos proveedores ofrecen servicios específicos que facilitan la gestión de bases de datos para los sistemas que se ejecutan en sus máquinas virtuales. Antes de desplegar esta base de datos, se deberá confirmar que el proveedor aplica redundancia en sus almacenamiento, garantizando así que si se produce un error en un disco y se pierde su contenido, la información se encuentre siempre duplicada y sea recuperable.

En el siguiente diagrama se encuentra representada la arquitectura física propuesta para el sistema:

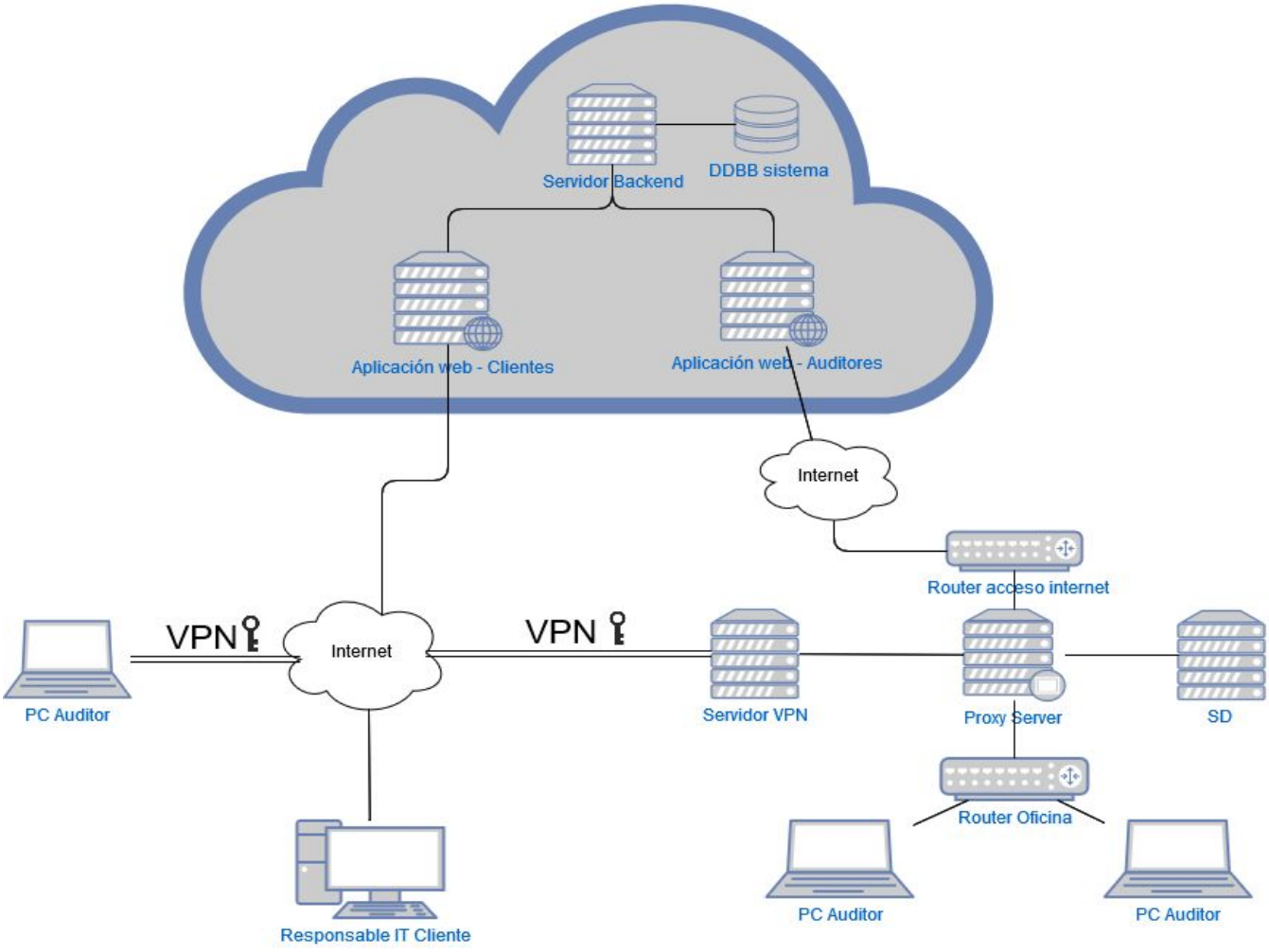


Figura 19. Diagrama de la arquitectura física del sistema

11.3.2.- Justificación

Se ha decidido apostar por esta distribución del hardware en el *cloud* debido a que tiene múltiples ventajas en comparación con la del despliegue actual del sistema. Las principales ventajas que ofrece se analizan a continuación:

- **Alta disponibilidad:** La gran mayoría de proveedores de servicios *cloud* ofrecen una disponibilidad de sus infraestructuras muy alta, entorno al 99,9% o superior según los SLAs (Service Level Agreement) de varios proveedores consultados [8] [9], por lo que se puede asegurar que por parte de estos proveedores la infraestructura estará disponible casi siempre para que los usuarios trabajen utilizando el sistema.

- **Facilidad en las actualizaciones:** Al encontrarse todo el código del sistema centralizado en estas máquinas virtuales, las actualizaciones del sistema son mucho más sencillas y se vuelven casi imperceptibles por los usuarios, ya que no deben de instalar ninguna actualización en sus dispositivos, sino que al acceder al sistema las actualizaciones ya se encuentran actualizadas. Esto también permite solucionar errores o realizar pequeñas mejoras de una forma mucho más ágil.

- **Seguridad:** Las infraestructuras de los grandes proveedores de servicios *cloud* disponen de las mejores medidas de seguridad, tanto física como lógica, para garantizar la seguridad de la información de los clientes que utilizan su infraestructura. Estos proveedores destinan muchos esfuerzos en garantizar esta seguridad, por lo que en muchos casos la información se encuentra más segura en su infraestructura que en los dispositivos y servidores de la compañía. Además, la organización del sistema en máquinas virtuales distintas para las aplicaciones de *frontend* y el servidor de *backend* proporcionan una capa extra de seguridad, ya que la información de los clientes se encuentra en el servidor y a este no puede acceder directamente ningún usuario a excepción de sus administradores.

  En el caso específico de las auditorías informáticas, el hecho de tener toda la información en el cloud y no en los dispositivos de los auditores también ofrece mucha más seguridad, debido a que si algún auditor perdiera su dispositivo y alguien consiguiera acceder a este, al no disponer de la información en local no podría acceder a ninguna información confidencial del cliente.

- **Escalabilidad:** El sistema está originalmente pensado para ser utilizado en las oficinas de la firma en España, pero en el caso de que su implementación sea altamente satisfactoria, cabría la posibilidad de poder compartir este software con las otras compañías de los distintos países en los que la red de de la firma tiene presencia.

Esta arquitectura permitiría desplegar el sistema fácilmente en las oficinas de otros países, ya que simplemente con copiar la misma configuración de las máquinas virtuales con el mismo proveedor *cloud* en las que se ejecutaría el sistema y luego configurar la red para que sólo se pueda acceder de forma interna sería suficiente para desplegar el sistema en otro país con nuevos usuarios y clientes.

- **Adaptabilidad:** Separar la lógica de negocio, en el servidor de *backend,* de la presentación del sistema, en las aplicaciones de *frontend,* permite que el sistema sea pueda adaptar a nuevas interfaces muy fácilmente en el futuro. Ya que si por ejemplo se quisiera desarrollar una aplicación móvil del sistema para los *smartphones* de los auditores, simplemente se debería desarrollar esta aplicación utilizando la interfaz ya existente del servidor para comunicarse con éste, de forma que los cambios a realizar en el servidor serían mínimos al poder adaptar perfectamente las APIs REST ya existentes en el servidor para su comunicación.

11.4- Tecnología a utilizar

La arquitectura definida, tanto lógica como física, es sólo un marco teórico que a la práctica se va a tener que implementar utilizando una tecnología concreta. Se considera que elegir un *stack* tecnológico adecuado que permita implementar esta arquitectura de la mejor forma posible es vital para el éxito de su desarrollo, ya que si se elige una tecnología que no esté alineada con esta arquitectura, la implementación del sistema será mucho más compleja o directamente no será posible.

Este apartado pretende dar solución a este problema proponiendo las tecnologías a utilizar para la implantación del sistema que se consideren más adecuadas teniendo en cuenta su arquitectura.

Aplicaciones web frontend

Para las aplicaciones web de *frontend*, se requiere que la tecnología a utilizar de soporte al desarrollo de aplicaciones web de tipo SPA (*Single Page Application*) y que se adapte al patrón arquitectónico MVC.

En los últimos años han ido apareciendo varios *frameworks* de *frontend* para el desarrollo de este tipo de aplicaciones web, un *framework* es una herramienta software que ofrece una forma estandarizada para construir y desplegar aplicaciones, pudiendo incluir programas adicionales, librerías y/o compiladores. Los más populares y utilizados por la comunidad en el desarrollo de *frontend* son: **Angular**, **React**, **Vue**, **Ember** y **Backbone**. Todos estos tienen en común que utilizan el lenguaje de programación Javascript y que soportan la aplicación del patrón arquitectónico MVC. [10]

Figura 20. Frameworks de *frontend* más populares[10]

No existe consenso en la comunidad sobre cual de estos *frameworks* es mejor, ya que cada uno tiene sus puntos fuertes en los que destaca, por lo que la decisión de cuál utilizar dependerá de detalles más técnicos y del equipo que se encargue de su desarrollo. Lo que sí se va a recomendar encarecidamente es la utilización de alguno de estos *frameworks* en lugar de desarrollar la aplicación desde cero, ya que estos ofrecen las siguientes ventajas:

- **Funcionalidades básicas:** Los *frameworks* llevan implementadas muchas funcionalidades básicas necesarias en el desarrollo de una aplicación web, por lo que tener que implementar todas estas funcionalidades básicas desde cero implicaría perder un tiempo que se podría haber dedicado a implementar las funcionalidades específicas de la aplicación. Además todas estas funcionalidades están fuertemente testadas, por lo que la probabilidad de que exista un *bug* en ellas es muy bajo.

- **Patrón MVC:** Al utilizar un *framework* el flujo de control viene dictado por el propio *framework* y no por el desarrollador. Al estar todos los *frameworks* basados en el patrón MVC, estos "obligaran" a los desarrolladores a seguir el patrón, de forma que la aplicación se desarrollará en línea con la arquitectura lógica propuesta.

- **Comunidad:** Todos estos *frameworks* poseen una comunidad de usuarios. En esta comunidad de usuarios algunos de ellos desarrollan módulos o extensiones para el *framework* y lo distribuyen gratuitamente. Por lo que si se necesita utilizar una funcionalidad que el *framework* no ofrece, pero existe alguien que la ha desarrollado y publicado en un módulo, se podría añadir esta funcionalidad al proyecto simplemente añadiendo el módulo y sin necesidad de desarrollarla.

Servidor de backend

Para el servidor de *backend,* se requiere que la tecnología a utilizar soporte el uso de APIs REST como interfaz con el propio servidor y que permita adaptar la arquitectura del servidor al patrón MVC.

Al igual que con la aplicación web, se deberá decidir si se considera necesario utilizar un *framework* o si se desarrollará todo el *backend* desde cero. Actualmente existen múltiples *frameworks* en el mercado que han sido diseñados específicamente para el desarrollo del *backend* de las aplicaciones web. Los más populares y utilizados por la comunidad son los siguientes: **Spring Boot**, **Ruby on Rails**, **ExpressJS**, **Django**, **Flask** y **Laravel**. Todos estos tienen en común que están basados en el patrón MVC, pero los lenguajes de programación difieren entre ellos. [11]

Al igual que con los *frameworks* de *frontend,* no existe un consenso en la comunidad sobre cual de estos *frameworks* es mejor, ya que cada uno tiene sus puntos fuertes en los que destaca, por lo que la decisión de cuál utilizar dependerá de detalles más técnicos, del lenguaje de programación que se quiera utilizar y del conocimiento de estos *frameworks* por parte del equipo que se encargue de su desarrollo.

El uso de estos *frameworks* tiene las mismas ventajas que las de los *frameworks* de *frontend*: llevan implementadas muchas funcionalidades básicas para el desarrollo del *backend* de una aplicación web y todas estas funcionalidades se encuentran fuertemente testeadas, se basan en el patrón MVC y tienen una gran comunidad de usuarios que desarrolla nuevos módulos con nuevas funcionalidades para añadir al *framework.* Por lo que en este caso también se recomienda encarecidamente el uso de uno de estos *frameworks.*

SO Máquinas virtuales

Los *frameworks* elegidos, tanto para el *frontend* como para el *backend,* se van a ejecutar en las máquinas virtuales del proveedor del *cloud*. Una máquina virtual no es más que un software capaz de emular un ordenador y que permite ejecutar un sistema operativo en su interior de la misma forma que lo hace un ordenador físico, de esta forma un solo dispositivo físico puede tener en su interior múltiples máquinas virtuales que ejecutan procesos y corren aplicaciones de la misma forma que lo haría un ordenador normal.

Las máquinas virtuales que se utilizarán para el sistema siguen este mismo principio, con la particularidad de que el servidor que las virtualiza se encuentra en el *cloud* del proveedor contratado. Para poder desplegar el sistema en ellas, primero necesitan disponer de un sistema operativo que se encargue de ejecutar el código desarrollado utilizando los *frameworks* mencionados, por lo que se tiene que decidir cuál será el SO que correrán estas máquinas.

Se ha observado que en la actualidad los sistemas operativos más utilizados para los servidores son Linux, utilizado en un 70,4% de servidores, y Windows Server, utilizado en otro 29,3%. Estos dos suman el 99.7% de servidores mundiales que los utilizan, por lo que el uso de otros sistemas operativos es totalmente residual. [12]

Debido a su popularidad en el uso, se recomienda utilizar uno de estos dos sistemas operativos, aún que la elección de cuál de los dos elegir no es evidente. Todos los *frameworks* comentados anteriormente, tanto para la aplicación de *frontend* como para el servidor de *backend,* son compatibles tanto con Linux o con WS, por lo que la elección final dependerá de detalles más técnicos, de las preferencias de los equipos encargados de su desarrollo, implementación y mantenimiento, y de las posibles licencias de las que ya disponga la compañía. (Por ejemplo, si ya se dispone de licencias de Windows Server, es posible que se prefiera utilizar este sistema operativo en lugar de Linux).

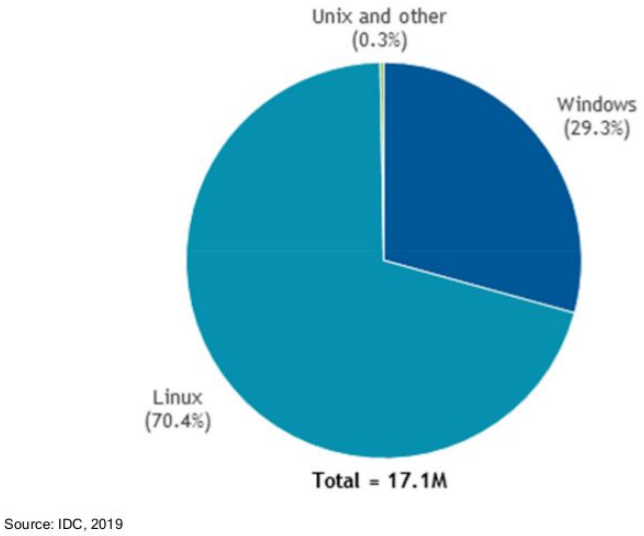


Figura 21. Cuota de mercado de los sistemas operativos utilizados en los servidores a nivel mundial [12]

12.- Diseño de la base de datos

En este apartado se pretende realizar una aproximación a cómo debería ser la estructura de una base de datos que permitiera generar la persistencia necesaria para el correcto funcionamiento del sistema especificado. El diseño realizado en este apartado es sólo una propuesta sobre la que se podría partir en el momento de realizar este diseño antes de ser implementada.

Para lograr el objetivo propuesto en este apartado, primero se va a obtener el diagrama de clases de diseño a partir del modelo de datos de especificación partiendo del modelo de datos de la fase de especificación, luego se analizará el problema a resolver y se decidirá qué tipo de BD a utilizar y por último, a partir de este modelo se formalizará una propuesta de esquema de BD en pseudocódigo, en la tecnología seleccionada, y se analizará si se pueden optimizar las consultas a esta BD de alguna forma, como por ejemplo con el uso de índices.

12.1.- Diagrama de clases de diseño

Para entender la información que se deberá persistir en la base de datos, serà de gran ayuda disponer del diagrama de clases del sistema adaptado para la fase de diseño, es decir, adaptando las relaciones entre clases para obtener un diagrama de clases que pueda ser representado en cualquier lenguaje de programación orientado a objetos. Además, en este diagrama se indicará la mínima navegabilidad entre clases necesaria para que el diseño e implementación de sus funcionalidades sea viable.

De esta forma se entenderá a alto nivel que conceptos deben de estar relacionados y hacia qué sentido irá la relación, ya que, por ejemplo, en la relación entre ITGC_concreto y Base no es lo mismo que el ITGC guarde una única referencia a la Base a la que pertenece, a que la Base guarde un listado con las referencias de todos los ITGCs concretos que tiene asignados.

A continuación se adjunta el diagrama de clases en UML adaptado a diseño, debido a que no se han generado nuevas restricciones textuales en este cambio, aplicarán las mismas que se definieron para la especificación.

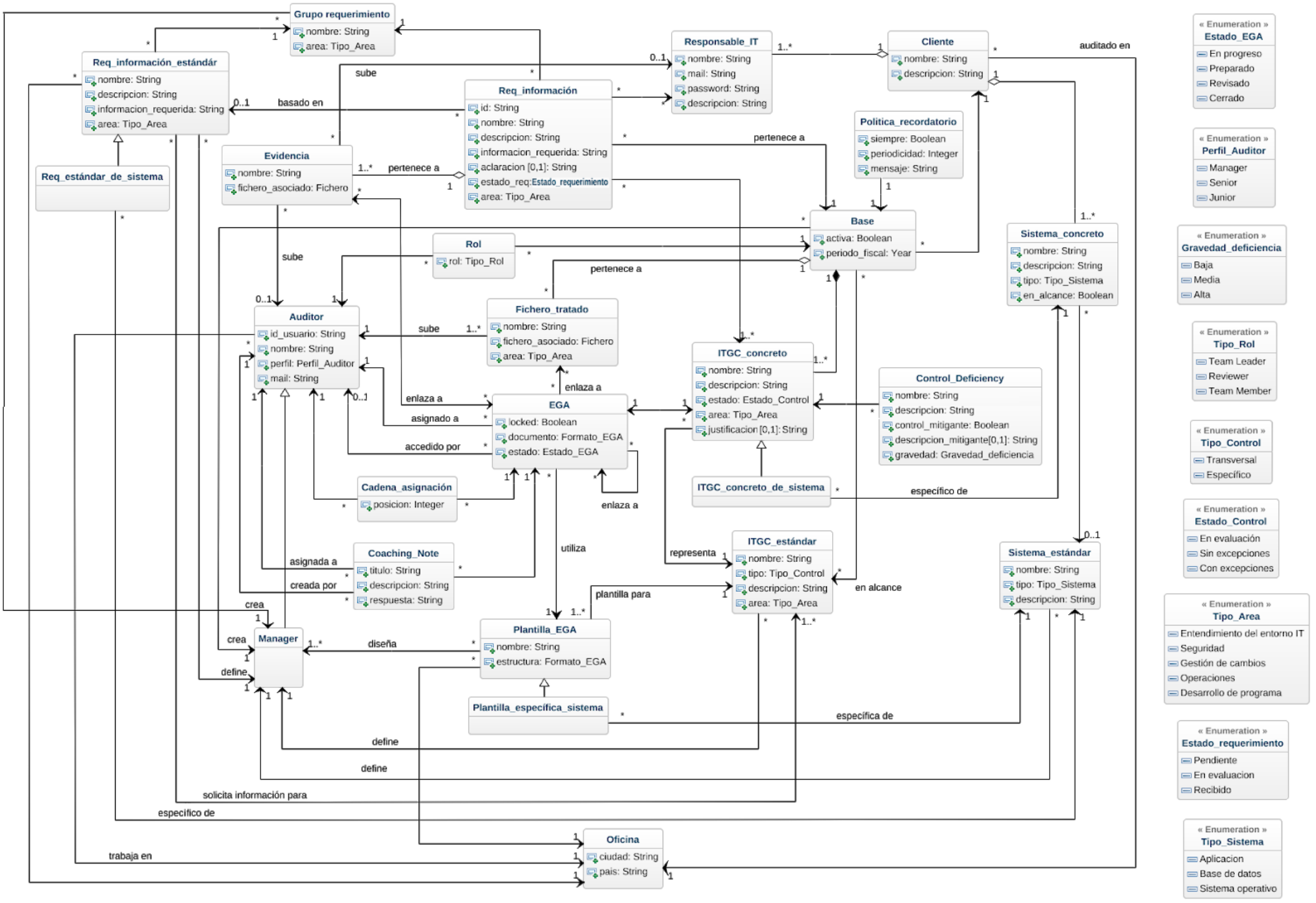


Figura 22. Diagrama de clases de diseño en UML

12.2.- Tecnología a utilizar

Antes de proceder con el diseño de la base de datos, es fundamental decidir, y justificar, la tecnología que se va a utilizar para mantener la persistencia, ya que dependiendo de la tecnología utilizada el diseño a realizar será distinto. Por ejemplo, no se diseña de la misma forma una base de datos relacional como una base de datos noSQL de tipo *key-value* o una base de datos noSQL de tipo grafo, por lo que será básico entender las necesidades de almacenamiento del sistema y decidir la tecnología que las pueda cubrir mejor.

Análisis de alternativas

En el mundo de las bases de datos, tenemos dos grandes famílias de BDs claramente diferenciadas: las bases de datos SQL, o relacionales, y las bases de datos noSQL, o no relacionales. Dentro de cada familia existen múltiples implementaciones, siendo la família de las noSQL mucho más diversa y con implementaciones muy distintas entre sí, pero lo que diferencia principalmente todas las implementaciones de bases de datos son los distintos modelos de consistencia de los datos que implementan.

Estos modelos de consistencia son conocidos por los acrónimos ACID y BASE, y no existe ningún consenso en la comunidad sobre cuál de los dos modelos es mejor, sino que cada modelo presenta sus ventajas e inconvenientes. A continuación se analizarán sus características.

Modelo ACID

Este modelo es el típicamente implementado por las bases de datos SQL o relacionales, la clave de este modelo es que provee de un entorno seguro para operar con los datos [13]. El acrónimo de ACID corresponde a:

- ***Atomic***: Asegura que o todas las operaciones en una transacción terminan con éxito o se hace *roll-back* de la operación (se vuelve al estado inicial revirtiendo los cambios introducidos por estas operaciones)

- ***Consistent****:* Al completar una transacción la base de datos queda en un estado consistente, de forma que todas las operaciones se han efectuado y los datos cumplen todas las restricciones establecidas sin inconsistencias entre ellos.

- ***Isolated***: Las transacciones son independientes entre ellas, de forma que ninguna transacción tiene acceso a otra transacción que no esté acabada. El acceso a los datos es moderado por la base de datos de forma que parece que las transacciones se ejecutan de forma sequencial.

- ***Durable:*** Los resultados de aplicar una transacción son permanentes y quedan registrados en los logs de estas transacciones.

Las propiedades ACID implican que cuando una transacción es completada, los datos en la BD siempre son consistentes, exactos, y con la última versión de estos estable en disco.

Modelo BASE

Este modelo apareció debido a que en las bases de datos tradicionales, las relacionales, existían dominios en los que las transacciones de tipo ACID aportaban un nivel de consistencia mucho más elevado del requerido, es decir, estas transacciones empleaban más recursos de los necesarios en garantizar la integridad de los datos en todo momento.

Es por eso que muchas de las bases noSQL han ido relajando los requisitos de consistencia y exactitud de las transacciones ACID para adoptar los principios BASE, de forma que se obtienen otros beneficios, como la escalabilidad o la resiliencia [13]. El acrónimo BASE corresponde a:

- ***Basic Availability:*** Garantiza la disponibilidad de los datos. Todas las solicitudes a la BD tendrán respuesta (que puede ser un error).

- ***Soft-state:*** El estado de una base de datos no tiene que ser siempre consistente, ni tampoco sus réplicas tienen que serlo, de forma que el estado del sistema puede ir variando en el tiempo.

- ***Eventual consistency:*** El sistema será consistente en algún punto en el futuro.

Se puede observar que las propiedades BASE son mucho menos estrictas que las ACID, principalmente en la consistencia de los datos, donde no es importante que el estado de la BD sea siempre consistente en cada instante sino que es suficiente que lo sea en el futuro. Esta relajación de las propiedades permite obtener una mejor disponibilidad, que a su vez permite una mejor escalabilidad del sistema, en detrimento de la seguridad e integridad que aporta la consistencia del modelo ACID.

No existe un modelo mejor o peor, ya que estos dos modelos son complementarios y dependiendo del dominio del sistema que se quiera implementar habrá un modelo que será más adecuado que el otro. Si se espera un gran volumen de usuarios, y este volumen está planificado que aumente en el tiempo o que se distribuya en distintas regiones geográficas, y no se considere importante disponer siempre de los datos más actualizados y sin incoherencias, se deberá priorizar la disponibilidad y escalabilidad proporcionada por el modelo BASE perdiendo parte de la consistencia de la BD como contrapartida. En el caso de que la prioridad sea la de tener siempre los datos más recientes y sin incoherencias en el sistema, y que el volumen de usuarios no sea muy elevado ni se espere un gran aumento de estos, se deberá priorizar la consistencia ofrecida por el modelo ACID obteniendo menos disponibilidad y escalabilidad como contrapartida.

Elección de la tecnología

Una vez comparados los dos modelos, se debe analizar las necesidades del dominio del sistema para decidir qué modelo, y dentro de este modelo la tecnología que lo implementa, son los más adecuados para cubrir las necesidades de este dominio.

Este sistema que se está diseñando está pensado para ser usado internamente por los auditores de la firma y puntualmente por los responsables de IT de los clientes que se conecten para subir evidencias, de forma que se espera una volumetría máxima de unos centenares de usuarios y no se prevé que esta cifra pueda aumentar con el tiempo, ya que el número de empleados del departamento acostumbra a ser bastante estable. Por lo que no se considera la escalabilidad del sistema como un factor importante y tampoco se prevé necesario que el sistema tenga una gran disponibilidad que le permita gestionar un gran volumen de solicitudes a la vez.

En cuanto a la consistencia de los datos, se considera muy importante que no existan incoherencias en la base que puedan inducir a errores y que la BD esté siempre en un estado consistente, ya que esto es debido a que un error en alguno de los elementos almacenados en la base podría impactar negativamente el dia a dia de los auditores. Además también se considera muy importante que todos los cambios y modificaciones que realicen los auditores queden debidamente registrados, que se preserve siempre la integridad de los datos y que existan *logs* que permitan trazar las transacciones realizadas.

Es por estos motivos que se considera el modelo de consistencia más adecuado para el sistema es el de transacciones tipo ACID, ya que ofrece una gran consistencia y nos asegura un entorno seguro para la realización de todas las operaciones. Además como la escalabilidad y la disponibilidad, los principales puntos fuertes del modelo BASE, no son factores prioritarios en este dominio, no se obtendría ninguna gran ventaja utilizando el modelo BASE, por lo que queda descartado.

Dentro del modelo ACID, encontramos principalmente las bases de datos relacionales y algunas bases de datos noSQL de tipo grafo [13]. Las bases de datos relacionales son las más típicas y más utilizadas por los desarrolladores, a diferencia de las bases de datos noSQL de tipo grafo, que son mucho más específicas para casos de uso concretos y su uso es mucho más residual [14]. Debido a que no existe ningún motivo específico para utilizar una BD de tipo grafo, se ha decidido diseñar una base de datos relacional, ya que esta tecnología es mucho más utilizada entre los desarrolladores y es compatible con un gran número de *frameworks* y lenguajes de programación distintos.

Dentro de las bases de datos relacionales existen múltiples implementaciones que se diferencian en pequeños detalles, pero en líneas generales todas utilizan variaciones de SQL como *query language* e implementan los mismos conceptos (tablas, filas, restricciones…), por lo que el diseño de la base de datos a realizar en este apartado no dependerá de la implementación específica de la BD elegida, sino que será genérico y aplicable a cualquier base de datos relacional que utilice SQL.

12.3.- Diseño de la base de datos

El diseño del esquema de la base de datos partirá del diagrama de clases de diseño y representará los elementos de este diagrama mediante tablas, columnas, restricciones e índices que podrán ser implementados en cualquier base de datos relacional que utilice SQL.

12.3.1.- Esquema de la base de datos

En este apartado se definirá el esquema de la base de datos, el cual contendrá toda la información necesaria para generar todas las tablas, con sus correspondientes campos, que darán soporte a la persistencia del sistema.

Para cada tabla se deberán especificar los siguientes elementos: nombre, que identificará la tabla; listado de campos, que representarán aquellos atributos o relaciones de la clase que deban ser persistidos; restricciones de integridad, que representarán parte de las restricciones textuales del diagrama; claves primarias y foráneas. Para cada atributo, se especificará el nombre del atributo, el tipo de datos que podrá contener, si es clave primaria y sus restricciones de integridad.

En el esquema de la base de datos, representado en pseudocódigo, se marcaran las claves primarias subrayando el campo, o campos, a los que corresponda la clave primaria; las claves foráneas se indicarán con la cláusula *ForeignKeys,* en la que se indicará la tabla y campo al que referencia cada campo de la tabla que es clave foránea; y por último, las restricciones de integridad se listaran al lado del campo que apliquen entre claudators {}, en caso de que apliquen a un solo campo, o dentro de la cláusula *Constraints,* en caso de que sean restricciones que apliquen a varios campos de la tabla.

El esquema de la base de datos se encuentra en el punto ***1.- Esquema de la base de datos del sistema en pseudocódigo*** del **Anexo.**

Este esquema se ha diseñado partiendo del diagrama de clases, los puntos más relevantes de cómo se han representado los elementos del diagrama son los siguientes:

- Respeto a las clases, cada una tiene una tabla asociada en la base de datos con el mismo nombre que la clase. Para todas estas clases (excepto para *Rol* y *Cadena asignación*, que representan relaciones entre otras clases y que no serán referenciadas por ninguna clase) se ha utilizado como *Primary Key* un identificador numérico que generará la base de datos automáticamente, de esta forma las relaciones entre clases se almacenan en la base de datos como *Foreign Keys* que referenciarán a estos identificadores.

- En cuanto a la herencia entre clases y subclases, se ha decidido que únicamente la tabla correspondiente a la superclase contenga los campos correspondientes a sus atributos y relaciones, que heredan también las subclases, y que luego las tablas de las subclases referencien a esta tabla. De forma que la tabla de cada subclase use como *Primary Key* una *Foreign Key* que referencie al identificador de su superclase y que solo contenga los campos correspondientes a los atributos o relaciones específicas de esta subclase.

- En cuanto a las relaciones entre clases:

    - Para aquellas relaciones con cardinalidad máxima de *uno* en alguno de los extremos se han almacenado estas relaciones utilizando *Foreign Keys* que referencian al identificador de la tabla con cardinalidad *uno* de la relación, en caso de que hubiera varias candidatas se ha decidido en qué tabla introducir la *Foreign Key* en función de la navegabilidad decidida en el diagrama.

    - Para aquellas relaciones con cardinalidades de *muchos a muchos,* al no poder almacenar un listado de valores en SQL, se ha visto la necesidad de utilizar tablas auxiliares que permitan representar estas relaciones. Estas tablas son las que su nombre sigue el patrón “Relacion_*Clase1*_*Clase2”* donde *Clase1* y *Clase2* corresponden a los nombres de las clases relacionadas. Para representar esta relación, estas tablas usan como *Primary Key* la combinación de las dos *Foreign Key* que referencian a los identificadores de las tablas correspondientes a la *Clase1* y *Clase2* respectivamente.

En el esquema resultante sólo se han tenido en cuenta las tablas y los campos necesarios para obtener un modelo funcional que pueda gestionar la persistencia del diagrama de clases de diseño, pero el aspecto de este esquema podría cambiar en el caso de que se añadieran *logs* de auditorías para registrar los accesos y las modificaciones en la base de datos, lo cual sería muy recomendable y necesario para cumplir con los requisitos no funcionales de *Seguridad.* En este apartado no se ha querido entrar en detalles sobre cómo se deberían registrar estos tipos de *logs*, por lo que se deja al equipo encargado de la implementación del sistema la responsabilidad de decidir cuál será la mejor implementación.

12.3.2.- Índices de las tablas

Los índices de las bases de datos relacionales son estructuras de datos asociadas a una tabla o vista que permiten acelerar las consultas de las filas de esa tabla o vista. Un índice contiene claves creadas a partir de una o más columnas de la tabla en concreto [15].

En las bases de datos relacionales, siempre se genera de forma automática un índice para la *Primary Key* de la tabla, esto es de gran utilidad ya que acelera la consulta necesaria para obtener los atributos correspondientes a un elemento concreto de la tabla. Por ejemplo, en el caso de que un elemento de la tabla referencie a un elemento de otra tabla mediante una *Foreign Key*, utilizando ese identificador, que a su vez es la *Primary Key* de esa otra tabla, se puede obtener muy rápidamente la fila correspondiente a ese identificador.

Aún así, habrá momentos en los que las consultas que se realizarán no corresponderán con la *Primary Key,* existen un gran número de combinaciones y de consultas distintas que se deberán implementar en la base de datos dependiendo del caso de uso y de la lógica de negocio involucrada con estos, pero para las consultas que se consideran más comunes es muy recomendable la creación de nuevos índices sobre las columnas de las tablas involucradas en estas consultas.
En este apartado se indicarán sobre qué columnas de las tablas de la base de datos se recomendaría introducir un índice para acelerar todas aquellas consultas que se puedan beneficiar de este. El motivo por el que se eligen estas columnas y no otras, es debido a que se considera que las consultas que las utilicen serán las más comunes y ejecutadas por la lógica de negocio del sistema.

Esta selección es sólo una recomendación, pues al implementar la base de datos es posible que se encuentren otras columnas más adecuadas para los índices que no se han tenido en consideración en este apartado. De la misma manera, tampoco se va a indicar que tipo de índice se deberá utilizar, ya que esto se deberá analizar y se deberá probar una vez las consultas esten bien definidas e implementadas en la base.

A continuación, se indicará para cada tabla sobre qué columnas se deberían implementar los índices y el motivo de esta decisión.

Tabla Responsable_IT

- **Columna *cliente*:** En el momento de asignar los requerimientos de información de una base a sus responsables, se tienen que obtener únicamente aquellos responsables que trabajen para el cliente auditado. Esto implica que el sistema deberá obtener aquellos responsables de IT cuyo campo *cliente* corresponda con el cliente que se audita en esa base concreta, por lo que introducir un índice sobre la columna *cliente* permitirá obtener más rápidamente los responsables que correspondan.

Tabla Req_informacion_estandar

- **Columna *oficina*:** Los requerimientos de información predeterminados considerados como estándares también dependen de la oficina para la que se esté auditando el cliente, por lo que sólo deberán aparecer como seleccionables aquellos requerimientos que correspondan a la oficina que audita el cliente. Esto implica que el sistema deberá obtener los requerimientos estándares que cumplan con este criterio, por lo que introducir un índice sobre la columna *oficina* permitirá obtener esta información de forma más eficiente.

Tabla ITGC_concreto

- **Columna *base*:** En el sistema, los ITGCs concretos siempre son accedidos por el auditor desde dentro de una base, es decir, siempre se accede a los ITGCs vía la base en la que se sitúa el auditor y no existe la posibilidad de acceder a estos sin pasar por la base. Esto implica que siempre que se acceda a una base se deberá consultar en la base de datos cuales son los ITGCs correspondientes a esa base, por lo que introducir un índice sobre la columna *base* de la tabla *ITGC_concreto* permitirá acelerar esa consulta, que se realizará siempre que un auditor abra una base.

Tabla Plantilla_EGA

- **Columna *ITGC_representado*:** Cuando se crea un EGA, el sistema tiene que mostrar las plantillas diseñadas para el ITGC estándar representado por el ITGC concreto de ese EGA. Esto implica que el sistema tiene que obtener de la base de datos los registros correspondientes a las plantillas asociadas a ese ITGC estándar, por lo que un índice sobre la columna *ITGC_representado* permitirá obtener más rápidamente aquellas plantillas correspondientes a un ITGC concreto.

- **Columna *oficina*:** El sistema no tiene que mostrar todas las plantillas asociadas al ITGC estándar, sino que debe filtrar y proporcionar sólo aquellas plantillas que además se correspondan a la oficina para la cual se está auditando el cliente. Esto implica que el sistema tiene que filtrar también por aquellas plantillas que correspondan a la oficina concreta, por lo que un índice sobre la columna *oficina,* más el índice ya existente sobre *ITGC_representado,* permitirá obtener los resultados que cumplan ambas condiciones de forma rápida y eficaz.

Tabla Req_informacion

- **Columna *base*:** Al igual que con los ITGCs concretos, los requerimientos de información son siempre específicos de una base y el auditor siempre accede a estos desde dentro de una base. Esto implica que cuando un auditor consulte el estado de los requerimientos de una base, se deberá realizar una consulta en la base de datos que devolverá aquellos registros correspondientes con estos requerimientos, por lo que introducir un índice sobre la columna *base* de la tabla *Req_informacion* permitirá acelerar en gran medida esta consulta.

Tabla Evidencia

- **Columna *req_informacion*:** La consulta de evidencias en una base se realiza siempre mediante los requerimientos de información, cada evidencia está siempre asociada a un único requerimiento y el auditor primero selecciona este requerimiento y luego el sistema le muestra las evidencias asociadas. Esto implica que cuando un auditor quiere consultar las evidencias asociadas a un requerimiento, el sistema tiene que obtener de la base de datos aquellos registros correspondientes a las evidencias asociadas a ese requerimiento, por lo que un índice sobre la columna *req_informacion* de la tabla *Evidencia* permitirá que esta consulta sea mucho más rápida y eficiente.

Tabla Fichero_tratado

- **Columna *base*:** Los ficheros tratados siempre son subidos por los auditores en una base concreta y también son consultados siempre para una base concreta, por lo que no es posible acceder a estos ficheros desde fuera de una base. Esto implica que cuando un auditor quiere consultar los ficheros tratados presentes en una base, el sistema tiene que obtener de la base de datos los registros correspondientes a los ficheros tratados asociados a esa base concreta, por lo que un índice sobre la columna *base* de la tabla *Fichero_tratado* permitirá acelerar esta consulta de forma que se podrá obtener más rápidamente los ficheros tratados de una base concreta.

Tabla EGA

- **Columna *itgc_concreto*:** Cada EGA se encuentra siempre asociado a un ITGC concreto de la base, cuando el auditor consulta los ITGCs concretos de una base tiene la opción de seleccionar el ITGC para que el sistema le muestre el EGA asociado. Esto implica que el sistema tiene que obtener de la base de datos el registro correspondiente al EGA que se encuentra asociado a ese ITGC concreto, por lo que un índice sobre la columna *itgc_concreto* permitirá realizar esa consulta de forma más rápida y eficiente.

- **Columna *auditor*:** Cuando un auditor selecciona la vista del *Dashboard*, el sistema le muestra todos aquellos EGAs pertenecientes a ITGCs de la base y que están asignados al auditor. Esto implica que el sistema tiene que obtener de la base de datos los registros correspondientes a los EGAs que se encuentran asociados a ITGCs concretos de esa base y al auditor concreto que quiere obtener la vista, por lo que un índice sobre la columna *auditor*, más el índice anterior sobre *itgc_concreto*, permitirá acelerar esta consulta en gran medida.

Tabla Auditor

- **Columna *oficina*:** Cuando un *Manager* quiere asignar auditores en una base concreta, el sistema tiene que filtrar entre todos los auditores de la compañía para mostrarle sólo aquellos auditores que trabajan en la misma oficina en la que se audita ese cliente. Esto implica que el sistema tiene que obtener los registros correspondientes a los auditores asignados a esa oficina, por lo que un índice sobre la columna *oficina* permitirá realizar esta búsqueda de una forma más rápida.

13.- Diseño de la interfaz

En este apartado se realizará una pequeña aproximación a cómo deberá ser la interfície gráfica del sistema para satisfacer correctamente las necesidades de los usuarios. Esta aproximación constará principalmente del diseño del mapa navegacional de las distintas pantallas con las que podrán interactuar los usuarios, además, para cada pantalla se dará una breve descripción de esta y se relacionará con aquellos casos de uso a los que se deberá tener acceso desde esa pantalla, especificando también que tipo de usuario tendrá acceso a la pantalla o a sus casos de uso concretos.

En este apartado no se tratará el diseño gráfico del sistema, pues se ha considerado que es más adecuado que este diseño sea realizado por alguien con más conocimientos y que esté más ligado con su futura implementación. Este apartado sólo pretende sentar las bases de la interfície gráfica del sistema, formalizando que pantallas se deberán crear y que casos de uso estarán asociados a cada pantalla, de esta forma se puede obtener una primera idea de cómo deberá ser esta interfaz y su relación con la lógica de negocio en cada pantalla.

Mapa navegacional - Auditores

Define las principales vistas de la interfaz correspondiente a la aplicación web con la que interactuarán los auditores, el mapa navegacional es el siguiente:

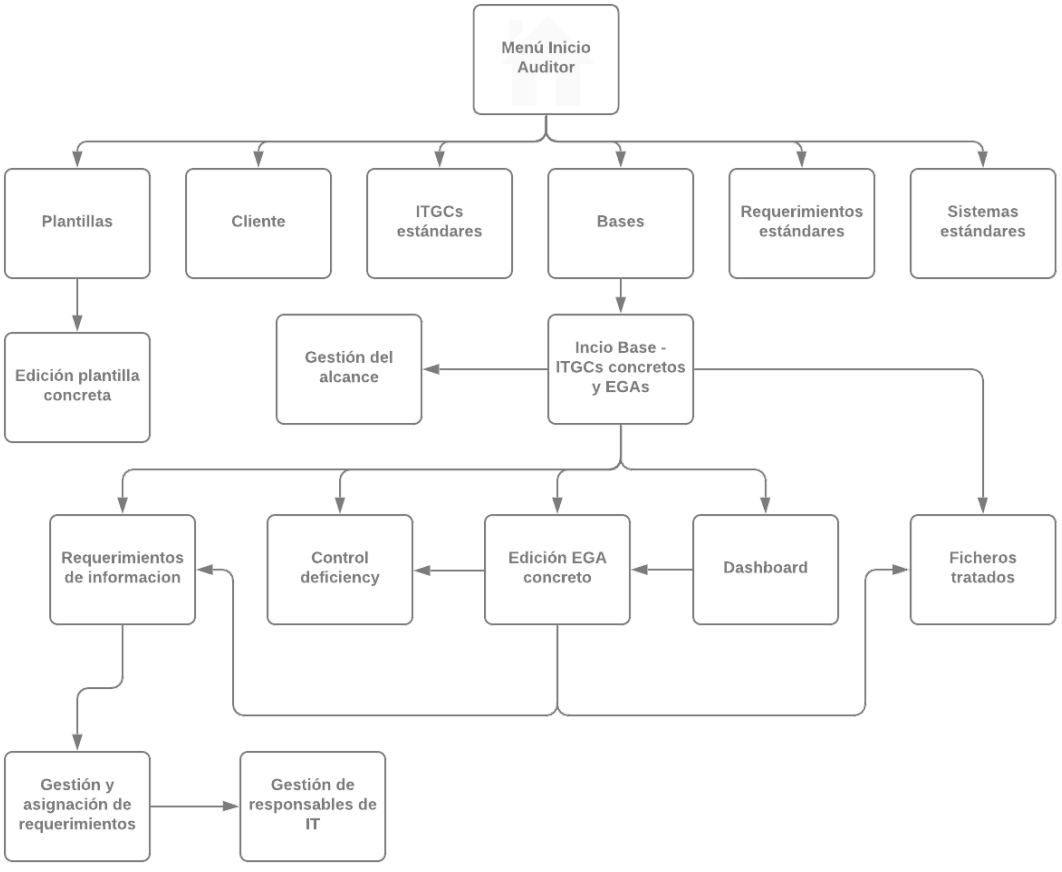


Figura 23. Mapa navegacional de la aplicación web para auditores

Cada uno de los elementos del mapa anterior corresponde a una pantalla principal y cada una de estas pantallas puede contener varias subpantallas, como por ejemplo distintos menús, pantallas de creación y/o edición de elementos…, pero para facilitar el análisi y la comprensión se agruparán todos estos elementos en una sóla pantalla y se relacionarán con las distintas funcionalidades definidas en la especificación que debe proveer el sistema.

A continuación se describen las pantallas que forman este mapa navegacional:

Menú Inicio Auditor

- **Usuarios con acceso:** Todos los auditores
- **Descripción:** Esta pantalla corresponde con el menú de inicio del sistema, en esta se deberá seleccionar la siguiente pantalla a la que se quiera navegar, teniendo en cuenta que la disponibilidad de las pantallas que aparecerán accesibles dependerá del tipo de usuario que acceda a esta. También se contempla que en esta pantalla se pueda configurar en un futuro elementos relacionados con la personalización y/o configuración del sistema.
- **Casos de uso relacionados:** El caso de uso *CU-48 - Acceder al sistema.*
- **Pantallas navegables:** Las pantallas son accesibles mediante links y dependiendo del tipo de auditor se pueden acceder a las siguientes:
    - Manager: *Plantillas, Sistemas estándares, Requerimientos estándares, ITGCs estándares y Bases*.
    - Auditores: *Bases.*

Sistemas estándares

- **Usuarios con acceso:** Managers.
- **Descripción:** En esta pantalla, los Managers pueden consultar los distintos sistemas que se consideran *estándares*, también pueden editar estos sistemas y crear un nuevo sistema si existe la necesidad. Estas dos últimas opciones podrán ser exclusivas de los managers responsables del departamento si la dirección considera necesario tener estos casos de uso más controlados.
- **Casos de uso relacionados:** Todos los casos de uso agrupados en la funcionalidad *1.-Gestión de sistemas estándares.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

ITGCs estándares

- **Usuarios con acceso:** Managers.
- **Descripción:** En esta pantalla, los Managers pueden consultar los distintos ITGCs que se consideran *estándares*, pueden editar la información de estos ITGCs y también crear un nuevos ITGCs estándares si existe la necesidad. Estas dos últimas opciones podrán ser exclusivas de los managers responsables del departamento si la dirección considera necesario tener estos casos de uso controlados.
- **Casos de uso relacionados:** Todos los casos de uso agrupados en la funcionalidad *2.-Gestión de ITGCs estándares.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

Plantillas estándares

- **Usuarios con acceso:** Managers
- **Descripción:** En esta pantalla, los Managers de una oficina concreta pueden consultar las plantillas disponibles para los distintos ITGCs estándares que se han diseñado en su oficina. Desde esta pantalla tendrán la opción de acceder al editor de aquellas plantillas que hayan sido creadas por el propio Manager, también podrán crear nuevas plantillas desde cero, copiar plantillas existentes o incluso eliminar sus propias plantillas.
- **Casos de uso relacionados:** Todos los casos de uso agrupados en la funcionalidad *3.-Gestión de plantillas.*
- **Pantallas navegables:** Desde esta pantalla se puede acceder a la pantalla *Edición plantilla concreta* al seleccionar la opción para editar una plantilla creada por el mismo Manager o se podrá volver la pantalla *Menú Inicio Auditor*.

Edición plantilla concreta

- **Usuarios con acceso:** Managers.
- **Descripción:** Esta pantalla permitirá dar forma a las plantillas que luego usarán los auditores para basar sus EGAs. En esta pantalla se deberá poder editar la estructura de la plantilla, por lo que será necesario que se puedan manipular todos sus elementos desde esta. En el sistema antiguo no existía el concepto de plantilla, sin embargo para editar los EGAs se utilizaba el programa *Microsoft Excel* juntamente con un *plug-in* desarrollado específicamente para el sistema, en el caso de que se siga la misma aproximación el trabajo a realizar en esta pantalla será mínimo, pero si se diseña un formato interno del sistema para representar a las plantillas será necesario proveer también de una pantalla que permita su correcta edición.
- **Casos de uso relacionados:** Todos los casos de uso agrupados en la funcionalidad *4.-Edición de plantillas.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

Requerimientos estándares

- **Usuarios con acceso:** Managers.
- **Descripción:** En esta pantalla, los Managers de una oficina concreta pueden consultar y gestionar los distintos requerimientos de información y los grupos de estos requerimientos que se consideran *estándares* en su oficina. Esto implica que en esta pantalla se pueden editar y/o eliminar los requerimientos estándares y los grupos de requerimientos ya existentes; y también se pueden crear nuevos requerimientos y grupos de requerimientos.
- **Casos de uso relacionados:** Todos los casos de uso agrupados en la funcionalidad *5.-Gestión de requerimientos de información estándares*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

Clientes

- **Usuarios con acceso:** Managers
- **Descripción:** En esta pantalla los managers podrán gestionar los clientes, juntamente con sus sistemas, que van a ser auditados en su oficina. En esta pantalla tendrán la opción de registrar nuevos clientes y editar la información de los clientes ya existentes, además para cada cliente podrán registrar nuevos sistemas, asociandolos con un sistema estándar en caso de que aplique; editar los sistemas ya existentes e incluso eliminar los sistemas que el cliente ya no utilice o no entren dentro del alcance de la auditoria.
- **Casos de uso relacionados:** Todos los casos de uso de las funcionalidades *6.-Gestión de clientes* y *9.-Gestión de sistemas de clientes.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

Bases

- **Usuarios con acceso:** Todos los auditores
- **Descripción:** En esta pantalla se mostrarán y gestionarán las bases del sistema. En esta pantalla los managers tendrán acceso a todas las bases del sistema, podrán crear nuevas bases en esta pantalla, asignar auditores de su oficina a una base y también editar el estado de una base que haya sido creada por ellos. En cambio, los auditores sólo tendrán acceso a la vista de *Mis bases,* en la que se mostrará el listado de bases que tiene asignadas un auditor concreto y permitirá seleccionar una de estas bases y abrirla, pasando así a una vista que contenga la información de esta base.
- **Casos de uso relacionados:** Todos los casos de uso de la funcionalidad *7.-Gestión de bases,* de la funcionalidad *8.-Gestión de auditores* y de la funcionalidad *18.-Acceso a las bases*
- **Pantallas navegables:** Desde esta pantalla, independientemente del rol del auditor, se podrá acceder a la pantalla *Menú base - ITGCs concretos y EGAs* o se podrá navegar de vuelta a la pantalla *Menú Inicio Auditor*.

Menú base - ITGCs concretos y EGAs

- **Usuarios con acceso:** Todos los auditores
- **Descripción:** Esta pantalla corresponde a la pantalla por defecto que se abrirá cuando los auditores seleccionen una base. En esta se mostrarán todos los ITGCs concretos de la base, agrupados por área de aplicación, y se podrá crear, acceder, modificar el estado y/o eliminar el EGA correspondiente a cada uno de estos. En esta pantalla existirán distintos tipos de vista (que presentarán la información de distintas formas) y existirán filtros para buscar entre todos los ITGCs.
- **Casos de uso relacionados:** Todos los casos de uso de las funcionalidades *23.-Administración de EGAs, 24.-Administración de la cadena de asignación, 28.-Filtrar por condiciones* y *30.- Administración de Coaching Notes*
- **Pantallas navegables:** Desde esta pantalla se podrá acceder a las pantallas *Gestión del alcance, Dashboard, Edición EGA concreto, Control Deficiency* y *Requerimientos de información.*

Gestión del alcance

- **Usuarios con acceso:** Team leaders
- **Descripción:** Esta pantalla será utilizada por el team leader para preparar la estructura de la base antes de los auditores empiecen a trabajar en ella. En esta pantalla el team leader podrá seleccionar los ITGCs estándares que se testearan en esta base, de forma que se generarán automáticamente los ITGCs concretos para esa base, también se podrán crear,editar y/o eliminar esos ITGCs de forma manual en esta pantalla. En caso de que aplique, también se podrán asignar sistemas concretos del cliente con el ITGC estándar, de forma que se generarán tantos ITGCs concretos de sistema como sistemas concretos del cliente se hayan asignado.
- **Casos de uso relacionados:** Todos los casos de uso de la funcionalidad *10.-Gestionar el alcance de la base*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs.*

Dashboard

- **Usuarios con acceso:** Todos los auditores.
- **Descripción:** En esta pantalla el auditor podrá consultar los EGAs de la base que tiene asignados en ese momento, también podrá consultar aquellos EGAs en los que se encuentre presente en su cadena de asignación (por ejemplo podrá ver que EGAs deberá revisar y en que estado se encuentran). En esta pantalla también se podrán gestionar las Coaching Notes de los EGAs, además de poder abrir los EGAs y modificar su estado.
- **Casos de uso relacionados:** Los casos de uso a los que se tendrá acceso en esta pantalla variarán en función del rol del auditor, algunos casos estarán disponibles para todos los auditores mientra que otros sólo serán accesibles para los reviewers o team leaders:
    - **Todos los auditores:** El *CU-69 Abrir EGA* de la funcionalidad *23.-Administración de EGAs* y todos los casos de uso de las funcionalidades *24.-Administración de la cadena de asignación, 27.-Preparar EGAs, 29.-Abrir el Dashboard* y *30.-Administración de Coaching Notes.*
    - **Reviewer:** El caso de uso de la funcionalidad *16. Revisión de EGAs*
    - **Team Leader:** El caso de uso de la funcionalidad *16.-Revisión de EGAs* y los casos de uso de la funcionalidad *11.-Cerrar EGAs.*
- **Pantallas navegables:** Desde esta pantalla se podrá acceder a la pantalla de *Edición EGA concreto* al seleccionar uno de los EGAs asignados y también se podrá navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs.*

Edición EGA concreto

- **Usuarios con acceso:** Todos los auditores.
- **Descripción:** Esta pantalla permitirá completar los EGAs en los que se documentará el trabajo de auditoría realizado para un ITGC concreto. En esta pantalla se deberá poder editar la estructura del EGA, por lo que será necesario que se puedan crear, editar y eliminar todos sus elementos desde esta. En el sistema antiguo se utilizaba el programa *Microsoft Excel* juntamente con un *plug-in* desarrollado específicamente para el sistema, en el caso de que se siga la misma aproximación el trabajo a realizar en esta pantalla será mínimo, pero si se diseña un formato interno del sistema para representar a las plantillas será necesario proveer también de una pantalla que permita su correcta edición.
- **Casos de uso relacionados:** Todos los casos de uso de la funcionalidad *25.-Edición de EGAs,* el caso de uso *CU-80 - Marcar EGA como preparado* y el caso de uso *CU-47 - Marcar EGA como revisado,* este último sólo en caso de que el auditor tenga el rol *reviewer* o *team leader*.
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs* o a la pantalla *Dashboard* (Dependiendo de la pantalla desde la que se haya accedido a esta). También se podrá navegar a las pantallas *Requerimientos de información* o *Ficheros tratados* para seleccionar las evidencias o ficheros tratados que se deberán linkar a un EGA concreto.

Control deficiency

- **Usuarios con acceso:** Todos los auditores.
- **Descripción:** En esta pantalla se podrán crear las “Control deficiencies” que se hayan detectado para un ITGC concreto, también se podrán modificar las deficiencias creadas anteriormente e incluso eliminarlas.
- **Casos de uso relacionados:** Todos los casos de uso de la funcionalidad *31.-Administración de Control Deficiencies*
- **Pantallas navegables:** Desde esta pantalla solo se podrá navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs.*

Requerimientos de información

- **Usuarios con acceso:** Todos los auditores.
- **Descripción:** En esta pantalla se mostrarán todos los requerimientos de información del sistema, agrupados por área de aplicación, los auditores podrán consultar cada uno de estos requerimientos y modificar su estado, además también podrán consultar las evidencias que se hayan subido para cada uno de estos y podrán administrar estas evidencias de forma manual.
- **Casos de uso relacionados:** Todos los casos de uso de las funcionalidades *19.-Clasificación de evidencias* y *20.-Administración manual de evidencias*
- **Pantallas navegables:** Los auditores con rol *reviewer* o *team leader* tendrán acceso a la pantalla de *Gestión y asignación de requerimientos* desde esta pantalla y todos los auditores podrán navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs.*

Gestión y asignación de requerimientos

- **Usuarios con acceso:** Auditores con el rol Reviewer o Team Leader.
- **Descripción:** En esta pantalla se gestionarán los requerimientos de información juntamente con los responsables de IT del cliente. En cuanto a los requerimientos, existirá la opción para generar automáticamente todos los requerimientos de información de la base teniendo en cuenta los ITGCs estándares en alcance y los requerimientos estándares, también existirá la opción de crear, editar y/o eliminar requerimientos de forma manual y asignarles a los responsables de IT del cliente los requerimientos creados que les correspondan.
- **Casos de uso relacionados:** Todos los casos de uso de las funcionalidades *13.-Gestión de requerimientos de información* y *14.-Asignación de requerimientos a sus responsables.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Requerimientos de información.*

Gestión de responsables de IT

- **Usuarios con acceso:** Auditores con el rol Reviewer o Team Leader.
- **Descripción:** En esta pantalla se gestionarán los responsables de IT del cliente y se dispondrá de la opción para enviarles un recordatorio, que abrirá una subpantalla desde la que se podrá personalizar el recordatorio.
- **Casos de uso relacionados:** Todos los casos de uso de las funcionalidades *12.-Administración de responsables de IT de los clientes,* y *15.- Gestión de recordatorios.*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Requerimientos de información.*

Ficheros tratados

- **Usuarios con acceso:** Todos los auditores.
- **Descripción:** En esta pantalla se podrán consultar los ficheros tratados que se encuentren presentes en la base, además de poder crear, editar y/o eliminar estos ficheros tratados de forma manual.
- **Casos de uso relacionados:** Todos los casos de uso de la funcionalidad *25.-Administración de ficheros tratados*
- **Pantallas navegables:** Desde esta pantalla solo se podrá navegar de vuelta a la pantalla *Menú Base - ITGCs concretos y EGAs.*

Mapa navegacional - Responsables IT

El mapa navegacional de la interfaz que utilizarán los responsables de IT para subir las evidencias para los requerimientos de información que tengan asociados es mucho más simple que el de la aplicación de los auditores, ya que sus funcionalidades son mucho más limitadas y se debe dar respuesta a un número muy inferior de casos de uso. Aún así, está interfaz es de vital importancia, ya que si no está bien diseñada e implementada se corre el riesgo de que los trabajadores del cliente no quieran utilizarla y prefieran seguir trabajando como se hace actualmente.

El mapa navegacional correspondiente a esta interfaz es el siguiente:

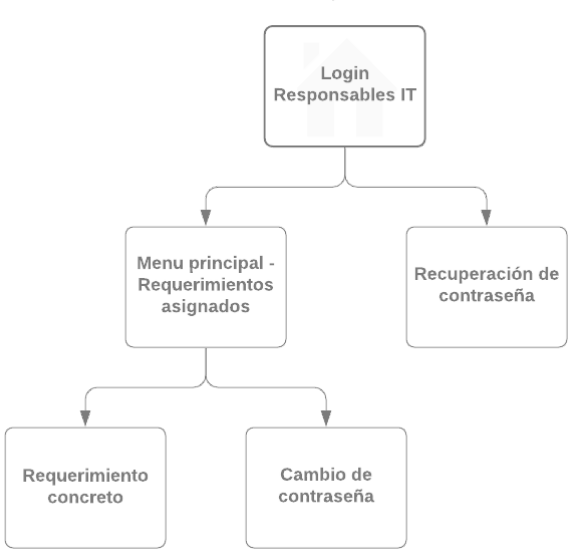


Figura 24. Mapa navegacional de la aplicación web para los responsables de IT del cliente

Cada uno de los elementos del mapa anterior corresponde a una pantalla principal y cada una de estas pantallas puede contener varias subpantallas, como por ejemplo distintos menús, pantallas de creación y/o edición de elementos…, pero para facilitar el análisi y la comprensión se agruparán todos estos elementos en una sóla pantalla y se relacionarán con las distintas funcionalidades definidas en la especificación que debe proveer el sistema.

A continuación se describen las pantallas que forman este mapa navegacional:

Login Responsables IT

- **Descripción:** En esta pantalla se identificarán los responsables de IT en orden de poder acceder al sistema. Para poder hacer *log in* deberán de introducir su correo corporativo juntamente con su contraseña, también tendrán la opción de solicitar la recuperación su contraseña en caso de que se olviden de esta.
- **Casos de uso relacionados:** Los casos de uso *CU-96 - Acceder al sistema* y *CU-98 - Recuperar contraseña*
- **Pantallas navegables:** Se podrá navegar a la pantalla *Menú principal - Requerimientos asignados*

Menú principal - Listado de requerimientos

- **Descripción:** En esta pantalla los usuarios podrán ver y filtrar los requerimientos de la base, por defecto se mostrarán los que estén asignados a ellos y en estado pendiente. Podrán abrir estos requerimientos para consultar su información y subir y/o eliminar evidencias.
- **Casos de uso relacionados:** Los casos de uso de la funcionalidad *33.- Consultar requerimientos de información.*
- **Pantallas navegables:** Se podrá navegar a las pantallas *Requerimiento concreto* y *Cambio de contraseña.*

Requerimiento concreto

- **Descripción:** En esta pantalla se podrán consultar los detalles de un requerimiento concreto y se podrán subir ficheros del dispositivo del usuario como evidencias asignadas a ese requerimiento. También se podrán eliminar las evidencias subidas por el propio usuario y/o descargar las evidencias subidas desde la aplicación web.
- **Casos de uso relacionados:** Los casos de uso de la funcionalidad *34.- Añadir y eliminar evidencias*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú principal - Requerimientos asignados.*

Cambio de contraseña

- **Descripción:** En esta pantalla los usuarios podrán cambiar su contraseña actual por una de nueva. Para hacerlo deberán introducir la actual y luego introducir dos veces la nueva contraseña y luego seleccionar el botón de cambiar.
- **Casos de uso relacionados:** El caso de uso *CU-97 - Cambiar contraseña*
- **Pantallas navegables:** Se podrá navegar de vuelta a la pantalla *Menú principal - Requerimientos asignados.*

Conclusiones

Una vez terminado con el proyecto, es momento de hacer retrospectiva y analizar cómo ha transcurrido el proyecto, si se han alcanzado los objetivos propuestos y si la solución aconseguida satisface las necesidades planteadas al principio del proyecto; de mirar hacia el futuro que tendrá este proyecto y también de reflexionar a nivel personal sobre la experiencia que ha supuesto.

En primer lugar, se considera que se han alcanzado los objetivos que se plantearon en el inicio del proyecto. Esta afirmación se fundamenta en que todos los objetivos de la fase de especificación se han cumplido, aunque con más trabajo de lo esperado, de forma satisfactoria y en que, aunque se redefinieron parte de los objetivos de la fase de diseño durante el transcurso del proyecto, al ser esta fase la menos prioritaria y teniendo en cuenta que los nuevos objetivos fueron del agrado de la dirección del departamento, se puede afirmar que los objetivos de la fase de diseño también se alcanzaron satisfactoriamente.

En segundo lugar, se considera que la solución obtenida cumple con unos buenos estándares de calidad y cubre las necesidades del departamento de auditoría de sistemas. Los estándares de calidad se han garantizado aplicando las metodologías y los conceptos aprendidos durante la especialidad de ES del GEI, ya que en ambas fases técnicas del proyecto se han aplicado conocimientos aprendidos en las distintas asignaturas del grado. Respecto a la correctez de la solución, se han validado los resultados obtenidos con la Sra. Andrea Bianchimano, gerente del departamento, y con varios auditores del mismo, además de que el propio estudiante ha participado en distintas auditorías de sistemas, por lo que se considera que estos elementos permiten afirmar que la solución planteada es correcta y cubre las necesidades del departamento.

De cara al futuro, este proyecto sólo tiene sentido si el sistema software termina siendo implementado. A día de hoy es bastante incierto el futuro que tendrá este sistema, ya que su implementación deberá ser autorizada por la dirección de la firma y en un escenario de pérdidas de ingresos es altamente probable que se recorte todo el gasto no esencial. Este escenario no podía ser previsto en el inicio del proyecto y deberá ser tratado por las direcciones del departamento y de la firma.

Por último, y a nivel personal, considero que este trabajo ha sido una buena experiencia que me ha permitido aplicar varios de los conceptos vistos en la facultad en una situación real. He podido comprobar de primera mano la complejidad que puede representar realizar la especificación completa de un sistema, la prueba de esto es que debido a la falta de experiencia previa en la especificación de sistemas software se subestimó en gran medida el coste en horas que supondría esta fase; y la importancia de que esta sea correcta, ya que a partir de la especificación se fundamentarán las otras fases que derivarán al desarrollo final del sistema. El proyecto también me ha permitido descubrir un nuevo sector laboral y me ha dado la oportunidad de poder aportar parte de mi trabajo al desarrollo de un nuevo sistema software que facilitará en un futuro el trabajo de los auditores de la firma.

Referencias

[1] ISACA. Glossary. En: *ISACA Resources*[En línea]. ISACA, 2020. [Consulta: 20 Febrero 2020]. Disponible en: https://www.isaca.org/resources/glossary

[2] Capterra. Best Audit Software. En: *Capterra Audit-Software*[En línea]. Capterra, 2020. [Consulta: 20 Febrero 2020]. Disponible en: https://www.capterra.com/audit-software

[3] Wikipedia. Servicio de directorio. En: *Wikipedia*[En línea]. Wikimedia foundation, 2020. [Consulta: 02 de Marzo de 2020]. Disponible en: https://es.wikipedia.org/wiki/Servicio_de_directorio

[4] Wikipedia. Servicio de directorio. En: *Wikipedia*[En línea]. Wikimedia foundation, 2020. [Consulta: 03 de Marzo de 2020]. Disponible en: https://es.wikipedia.org/wiki/Red_privada_virtual#cite_note-1

[5] International Organization for Standardization (ISO). ISO/IEC 25000:2014(en). En: *Online Browsing Platform (OBP)*[En línea]. ISO/IEC, 2014. [Consulta: 20 de Abril de 2020]. Disponible en:https://www.iso.org/obp/ui/#iso:std:iso-iec:25000:ed-2:v1:en

[6] ISO/IEC 25010. En: *NORMAS ISO 25000*[En línea]. iso25000.com, 2019. [Consulta: 20 de Abril de 2020]. Disponible en: https://iso25000.com/index.php/normas-iso-25000/iso-25010

[7] Harshal Patil. Contemporary Front-end Architectures. En: *Medium - <web/F>*[En línea]. Medium Corporation, 05 de Septiembre de 2019. [Consulta: 29 de Abril de 2020]. Disponible en: https://blog.webf.zone/contemporary-front-end-architectures-fb5b500b0231

[8] Amazon. Contrato de nivel de Servicios de Computación de Amazon. En: *Amazon AWS*[En línea]. Amazon Web Services Inc, 2020. [Consulta el 04 de Mayo de 2020]. Disponible en: https://aws.amazon.com/es/compute/sla/

[9] Microsoft. Resumen de SLA para los servicios de Azure. En: *Microsoft Azure*[En línea]. Microsoft, 2020. [Consulta: el 04 de Mayo de 2020]. Disponible en: https://azure.microsoft.com/es-es/support/legal/sla/summary/

[10] Sviatoslav A. The Best JS Frameworks for Front End. En: *RubyGarage - Blog*[En línea]. RubyGarage, 8 de Enero de 2020. [Consulta: 06 de Mayo de 2020]. Disponible en: https://rubygarage.org/blog/best-javascript-frameworks-for-front-end

[11] Sushil Kumar. Top 7 Backend Web Development Frameworks 2019. En: *KelltonTech*[En línea]. Kellton Tech, 11 de Marzo de 2019. [Consulta: 07 de Mayo de 2020]. Disponible en: https://www.kelltontech.com/kellton-tech-blog/top-7-backend-web-development-frameworks-

2019

[12] The Red Hat Enterprise Linux Team. Red Hat: Leading the enterprise Linux server market. En: *Blog de Red Hat*[En línea]. Red Hat Inc, 2020. [Consulta: 09 de Mayo de 2020] Disponible en:https://www.redhat.com/es/blog/red-hat-leading-enterprise-linux-server-market

[13] Bryce Merkl. Graph Databases for Beginners: ACID vs. BASE Explained. En: *neo4j - blog*[En línea]. Neo4j Inc, 13 de Noviembre de 2018. [Consulta: 12 de Mayo de 2020] Disponible en: https://neo4j.com/blog/acid-vs-base-consistency-models-explained/

[14] ScaleGrid. 2019 Database Trends. En: *ScaleGrid - Blog*[En línea] ScaleGrid, 04 de Marzo de 2020. [Consulta: 14 de Mayo de 2020] Disponible en: https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/

[15] Microsoft. Clustered and Nonclustered Indexes Described. En *SQL Docs*[En línea]. 2020 Microsoft. [Consulta: 27 de Mayo de 2020] Disponible en: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15

Anexo

1.- Tabla de tareas

Fase	Sub-fase	Tarea	Tiempo (h)	Dependiente de	Recursos específicos
Especificación	Estudio de la situación y sistema actual	T1 - Estudiar el contexto	6	-	Procesador de texto
		T2 - Modelizar procesos	20	T1	Procesador de texto + programa para modelar
		T3 - Identificar puntos fuertes y débiles	4	T2	Procesador de texto
		T4 - Visión de proyecto, oportunidades de mejora y alternativas	8	T3	Procesador de texto
		T5 - Identificar los stakeholders	4	T4	Procesador de texto
		T6 - Identificar los objetivos a cumplir	4	T4	Procesador de texto
		T7 - Tecnología implicada y sistemas a interaccionar	4	T4	Procesador de texto
		T8 - Validar tareas	1	T7	Andrea Bianchimano
		T9 - Reunión de seguimiento	1	T7	Ponente TFG
		T10 - Documentar las tareas anteriores	2	T8	Procesador de texto
	Análisis y especificación de requisitos	T11.1 - Identificar los requisitos funcionales	12	T10	Procesador de texto
		T11.2 - Identificar los requisitos no funcionales	2	T10	Procesador de texto
		T12 - Validar los requisitos definidos	2	T11	Equipo de SPA
		T13 - Identificar, analizar y definir los casos de uso del sistema	20	T12	Procesador de texto
		T14 - Validar los casos de uso	1	T13	Andrea Bianchimano
		T15 - Reunión de seguimiento	1	T13	Ponente TFG
		T16 - Documentar las tareas anteriores	2	T14	Procesador de texto
	Especificación del sistema en un diagrama de clases	T17 - Identificar las clases del dominio	2	T16	-
		T18 - Definir las relaciones entre clases	2	T17	-
		T19 - Plasmar las clases y las relaciones en un diagrama UML	2	T18	Programa para modelar en UML
		T20 - Definir las restricciones textuales	2	T19	Procesador de texto
		T21 - Validar el diagrama de clases	1	T20	Andrea Bianchimano
		T22 - Reunión de seguimiento	1	T20	Ponente TFG
Diseño	Diseño de la arquitectura del sistema	T23 - Definir la arquitectura física	2	T20	Procesador de texto + programa para modelar
		T24 - Definir diagrama de clases de diseño	1	T20	Programa para modelar en UML
		T25 - Identificar los atributos y las funciones necesarias	20	T24	Procesador de texto
		T26 - Identificar patrones a utilizar	6	T24	Procesador de texto
		T27 - Reunión de seguimiento	1	T26	Ponente TFG
		T28 - Documentar las tareas anteriores	2	T26	Procesador de texto
	Diseño de la estructura de la BD	T29 - Analizar y decidir el tipo de BD a utilizar.	2	T23	Procesador de texto
		T30 - Identificar las tablas necesarias	1	T24	-
		T31 - Definir los campos	2	T25	Procesador de texto
		T32 - Definir las restricciones necesarias	2	T31	Procesador de texto
		T33 - Definir las relaciones entre tablas	4	T32	Procesador de texto + programa para modelar
		T34 - Identificar posibles índices a utilizar	3	T33	Procesador de texto
		T35 - Reunión de seguimiento	1	T34	Ponente TFG
		T36 - Documentar las tareas anteriores	2	T35	Procesador de texto
	Diseño de la interfaz del sistema	T37 - Diseñar mock-ups de las vistas principales	20	T24	Programa para diseño mock-ups
		T38 - Definir mapas navegacionales	20	T37	Procesador de texto + programa para modelar
		T39 - Reunión de seguimiento	1	T38	Ponente TFG
		T40 - Documentar las tareas anteriores de cara a la entrega final	5	T39	Procesador de texto
TOTAL	-	**Suma de tareas**	**199**	-	-

Figura 25. Tabla de tareas planificadas en el inicio del proyecto

2 - Diagrama de Gantt

Fase especificación

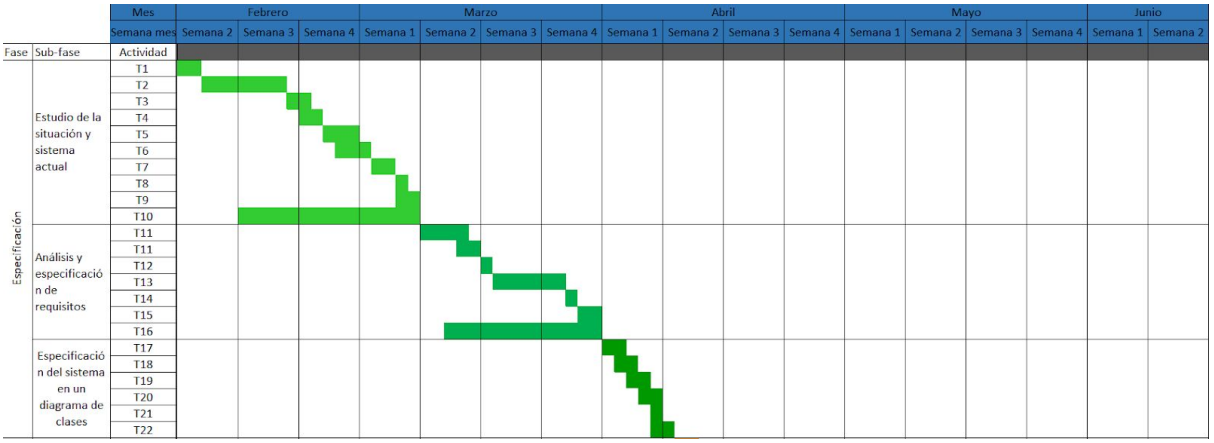


Figura 26. Diagrama de Gantt de la fase de especificación

Fase diseño

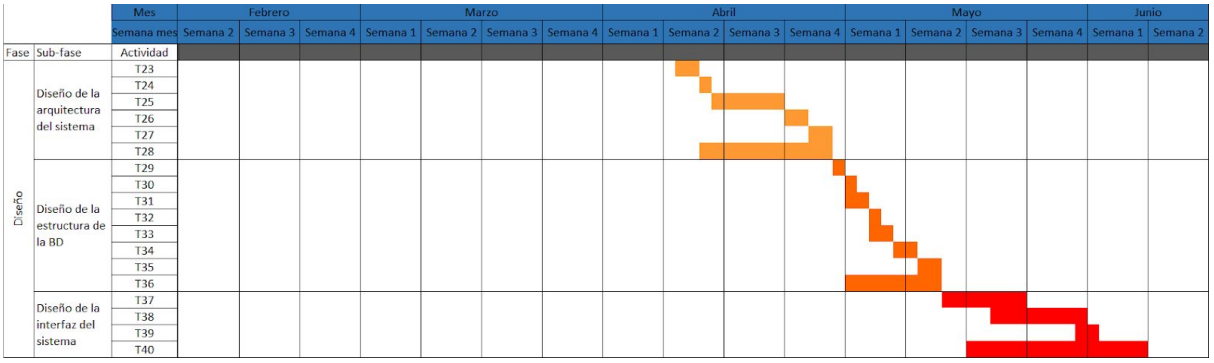


Figura 27. Diagrama de Gantt de la fase de diseño

3.- Especificación completa de los casos de uso del sistema

Auditor - Manager

Los casos de uso del actor correspondiente al auditor con perfil *Manager* se han analizado en 2 grupos formados por las distintas funcionalidades a las que tiene acceso este perfil.

El primer grupo, denominado *Gestión de elementos predefinidos,* agrupa las funcionalidades correspondientes a la gestión de los elementos estándares del sistema y está formado por las funcionalidades siguientes: *1.- Gestión de sistemas estándares, 2.- Gestión de ITGC estándares, 3.- Gestión de plantillas, 4.- Edición de plantillas y 5.- Gestión de requerimientos de información estándares.*

El segundo grupo, denominado *Gestión de bases,* agrupa las funcionalidades relacionadas con la administración de las bases y está formado por las funcionalidades siguientes: *6.- Gestión de clientes, 7.- Administración de bases, 8.- Gestión de auditores* y *9.- Gestión de sistemas de clientes.*

Gestión de elementos predefinidos

Este conjunto de funcionalidades permitirá gestionar los elementos estándares que los gerentes habrán definido antes de la realización de las auditorías. Dentro de estos elementos se encontrarán ITGCs, sistemas, req. de información y plantillas para EGAs.

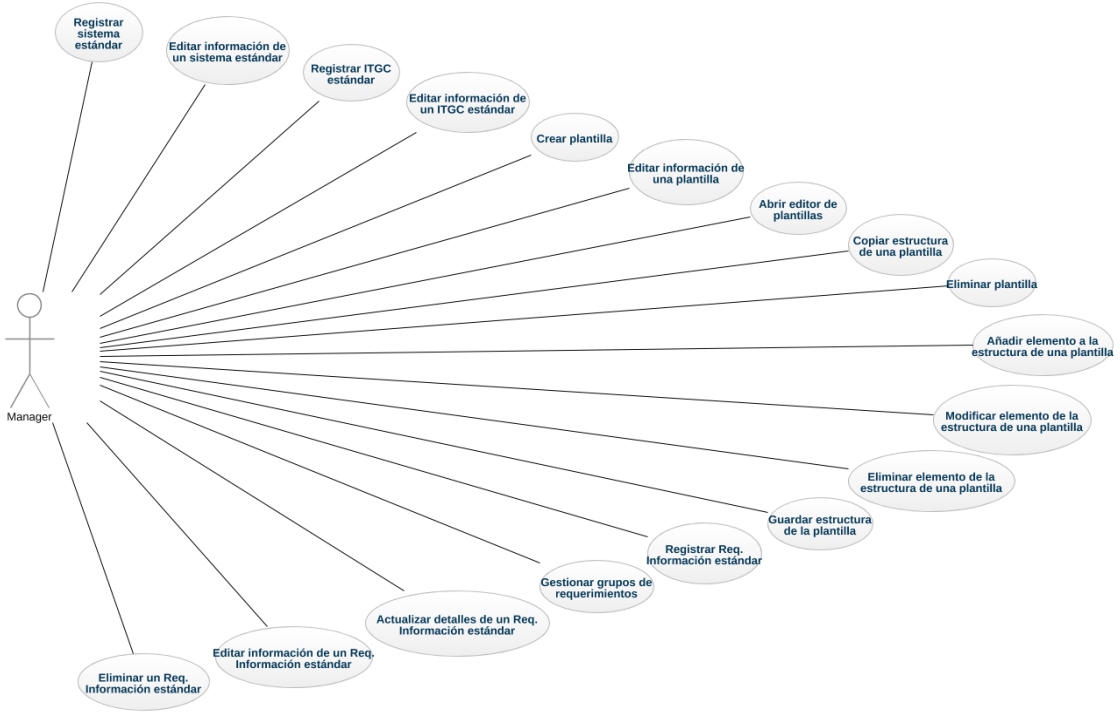


Figura 28. Diagrama de casos de uso para la gestión de elementos predefinidos

1.- Gestión de sistemas estándares

- **CU-01 - Registrar sistema estándar:**
  - El usuario selecciona la opción “Sistemas estándares” y luego selecciona la opción “Nuevo sistema”.
  - El sistema abre una ventana donde el usuario introduce el nombre del sistema y opcionalmente una descripción.
  - En el caso de que no exista ningún sistema estándar registrado con el mismo nombre: el sistema crea y almacena el sistema estándar.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

- **CU-02 - Editar información de un sistema estándar:**
  - El usuario selecciona la opción “Sistemas estándares” y el sistema le muestra un listado con todos los sistemas registrados.
  - El usuario selecciona el sistema estándar a editar. y el sistema abre una ventana con la información del sistema estándar.
  - El usuario modifica el nombre y/o la descripción y una vez terminado selecciona la opción “Guardar”.
  - En el caso de que no exista ningún sistema estándar registrado con el mismo nombre: el sistema actualiza los campos del sistema estándar.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

2.- Gestión de ITGCs estándares

- **CU-03 - Registrar ITGC estándar:**
  - El usuario selecciona la opción “ITGCs estándares” y luego selecciona la opción “Nuevo ITGC estándar”.
  - El sistema abre una ventana donde el usuario introduce el nombre del ITGC, selecciona el área de aplicación que le corresponde, selecciona si el control es transversal o específico de sistema y opcionalmente introduce una descripción.
  - En el caso de que no exista ningún ITGC estándar en el sistema con el mismo nombre: el sistema crea y almacena el ITGC estándar.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

- **CU-04 - Editar información de un ITGC estándar:**
  - El usuario selecciona la opción “ITGCs estándares” y el sistema le muestra un listado con todos los ITGC registrados agrupados por área de aplicación.
  - El usuario selecciona un área de aplicación y el sistema despliega el listado con todos los ITGCs de esa área.
  - El usuario selecciona el ITGC del listado y el sistema abre una ventana con la información del ITGC estándar.
  - El usuario modifica el nombre, la descripción, el tipo de control y/o el área de aplicación, una vez terminado selecciona la opción “Guardar”.
  - En el caso de que no exista ningún ITGC estándar en el sistema con el mismo nombre: el sistema guarda los cambios en el ITGC estándar.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

3.- Gestión de plantillas

- **CU-05 - Crear plantilla:**
  - El usuario selecciona la opción “Plantillas” en el menú principal y luego selecciona la opción “Nueva plantilla”.
  - El sistema abre una ventana donde el usuario introduce el nombre de la plantilla y selecciona el ITGC estándar, de un listado con todos los ITGCs registrados agrupados por área de aplicación, para el que servirá como plantilla.
  - Si el ITGC seleccionado es de tipo específico de sistema:
    - El sistema añade la opción para marcar la plantilla como específica de sistema.
    - Si el usuario selecciona la opción, este debe también seleccionar el sistema estándar, de un listado con todos los sistemas estándares registrados, para el que aplicará.
  - En caso de que no exista una plantilla con el mismo nombre para ese ITGC estándar: el sistema crea y almacena la nueva plantilla.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

- **CU-06 - Editar información de una plantilla:**
  - El usuario selecciona la opción “Plantillas” y el sistema le muestra un listado con todas las plantillas disponibles agrupadas por ITGC estándar, que a su vez están agrupados por área de aplicación.
  - El usuario selecciona la plantilla, luego selecciona la opción “Editar” y sistema abre una ventana con la información de la plantilla.
  - El usuario modifica el nombre, el ITGC asociado y/o el tipo de plantilla.
  - En caso que el usuario selecciona el tipo “Específica”:
    - El usuario selecciona el sistema estándar para el que aplicará de un listado con todos los sistemas estándares registrados.
  - Una vez terminado selecciona la opción “Guardar”.
  - En caso de que no exista una plantilla con el mismo nombre para ese ITGC: el sistema actualiza la información de la plantilla.
  - En el caso contrario: el sistema muestra un mensaje informando del error.

- **CU-07 - Abrir editor de plantillas:**
  - El usuario selecciona la opción “Plantillas” y el sistema le muestra un listado con todas las plantillas disponibles agrupadas por ITGC estándar, que a su vez están agrupados por área de aplicación.
  - El usuario selecciona la plantilla de este listado y el sistema abre una ventana en la que puede editar la estructura de esa plantilla concreta.

- **CU-08 - Copiar estructura de una plantilla:**
  - El usuario selecciona la opción “Plantillas” y el sistema le muestra un listado con todas las plantillas disponibles agrupadas por ITGC estándar, que a su vez están agrupados por área de aplicación.
  - El usuario selecciona la plantilla de este listado, luego selecciona la opción “Copiar estructura” y el sistema le abre una ventana con el listado de todas las plantillas disponibles.
  - El usuario selecciona la plantilla de la que quiere copiar su estructura y selecciona la opción “Copiar”,
  - El sistema modifica la estructura de la plantilla para que pase a ser la misma que la de la plantilla que se ha copiado.

- **CU-09 - Eliminar plantilla:**
  - El usuario selecciona la opción “Plantillas” y el sistema le muestra un listado con todas las plantillas disponibles agrupadas por ITGC estándar, que a su vez están agrupados por área de aplicación.
  - El usuario selecciona la plantilla de este listado y luego selecciona la opción “Eliminar plantilla”.
  - El sistema abre una ventana para confirmar la operación y el usuario selecciona “Eliminar”.
  - El sistema elimina la plantilla.

4.- Edición de plantillas

El usuario accede a los siguientes casos de uso a partir del caso CU-07.

- **CU-10 - Añadir elemento a la estructura de una plantilla:**
  - El usuario selecciona el contenedor del elemento que va a añadir a la estructura, excepto si el elemento es una hoja, que en este caso se añadirá directamente.
  - El sistema muestra el listado de elementos a añadir.
  - El usuario elige el elemento deseado.
  - El sistema crea este elemento seleccionado dentro de su contenedor.

- **CU-11 - Modificar elemento de la estructura de una plantilla:**
  - El usuario selecciona un elemento de la estructura.
  - El sistema muestra todas las opciones para la modificación del elemento.
  - El usuario selecciona la opción y modifica el elemento.

- **CU-12 - Eliminar elemento de la estructura de una plantilla:**
  - El usuario selecciona un elemento de la estructura y luego selecciona la opción “Eliminar”.
  - El sistema elimina el elemento de la estructura.

- **CU-13 - Guardar estructura de una plantilla:**
  - El usuario selecciona la opción “Guardar”.
  - El sistema guarda la estructura de la plantilla teniendo en cuenta todos los elementos que han sido añadidos, modificados y/o eliminados.

5.- Gestión de requerimientos de información estándares

- **CU-14 - Gestionar grupos de requerimientos**
  - El usuario selecciona la opción “Grupos de requerimientos” y el sistema muestra un listado con todos los grupos de requerimientos registrados.
  - El usuario puede:
    - Seleccionar la opción “Nuevo grupo”, donde el sistema abrirá una ventana para que el usuario introduzca el nombre y el área de aplicación del grupo.
    - Seleccionar un requerimiento:
      - Seleccionar la opción “Editar”, el sistema abrirá una ventana en la que el usuario podrá modificar el nombre del grupo.
      - Seleccionar la opción “Eliminar”, el sistema solicitará la confirmación de la operación y si el grupo no tiene ningún requerimiento asignado lo eliminará.
    - En el caso de creaciones y ediciones, si no existe un grupo de esa área de aplicación con el mismo nombre el sistema la registrará, en caso contrario mostrará un mensaje informando del error.

- **CU-15 - Registrar req. información estándar:**
  - El usuario selecciona la opción “Requerimientos de Información estándares” y luego selecciona la opción “Nuevo Requerimiento estándar”.
  - El sistema abre una nueva ventana en la que el usuario introduce el nombre del requerimiento, selecciona el área de aplicación al que corresponde y selecciona los ITGCs estándares, de un listado con todos los controles correspondientes al área seleccionada, para los que el requerimiento solicita evidencias. Opcionalmente introduce la descripción del requerimiento y el detalle de la información requerida.
  - En caso de que los ITGCs seleccionados sean todos de tipo específico de sistema:
    - El sistema permite marcar el requerimiento como específico de sistema y seleccionar el sistema estándar que le corresponda de un listado con todos los sistemas estándares registrados.
  - En el caso de que no exista un requerimiento estándar registrado con el mismo nombre, el sistema crea y almacena el requerimiento.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-16 - Actualizar detalles de un req. información estándar:**
  - El usuario selecciona la opción “Requerimientos de Información estándares” y el sistema muestra un listado con todos los requerimientos estándares registrados agrupados por su área de aplicación.
  - El usuario selecciona un requerimiento y el sistema abre una ventana con la descripción del requerimiento y el detalle de la información requerida.
  - El usuario modifica estos campos y una vez ha terminado selecciona la opción “Guardar”.
  - El sistema almacena la actualiza los cambios en los detalles del requerimiento.

- **CU-17 - Editar información de un req. información estándar:**
  - El usuario selecciona la opción “Requerimientos de Información estándares” y el sistema muestra un listado con los requerimientos estándares registrados agrupados por su área de aplicación.
  - El usuario selecciona un requerimiento y luego selecciona la opción “Editar Información”
  - El sistema muestra una ventana con el nombre del sistema, su área y un listado con los controles que tiene asociados.
  - El usuario puede modificar el nombre, seleccionar otra área (de forma que todos los controles q tenia asociados se eliminan), eliminar un control eligiendolo del listado y seleccionando la opción “Eliminar” y/o añadir controles a partir del listado de ITGCs estándares correspondientes al área del requerimiento.
  - En el caso de un requerimiento específico:
    - El usuario puede cambiar el sistema que tiene asignado a partir de un listado con todos los sistemas estándares registrados.
  - Una vez finalizado el usuario selecciona la opción “Guardar”.
  - En caso de que no exista un requerimiento con el mismo nombre el sistema actualiza la información del requerimiento.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-18 - Eliminar un req. información estándar:**
  - El usuario selecciona la opción “Requerimientos de Información estándares” y el sistema muestra un listado con los requerimientos estándares registrados agrupados por su área de aplicación.
  - El usuario selecciona un requerimiento y luego selecciona la opción “Eliminar”.
  - El sistema abre una ventana para confirmar la acción y el usuario selecciona “Eliminar”.
  - El sistema elimina el requerimiento de información estándar..

Gestión de bases

Los siguientes casos de uso permitirán gestionar las bases de auditoría juntamente con los conceptos y funcionalidades relacionados con estas.

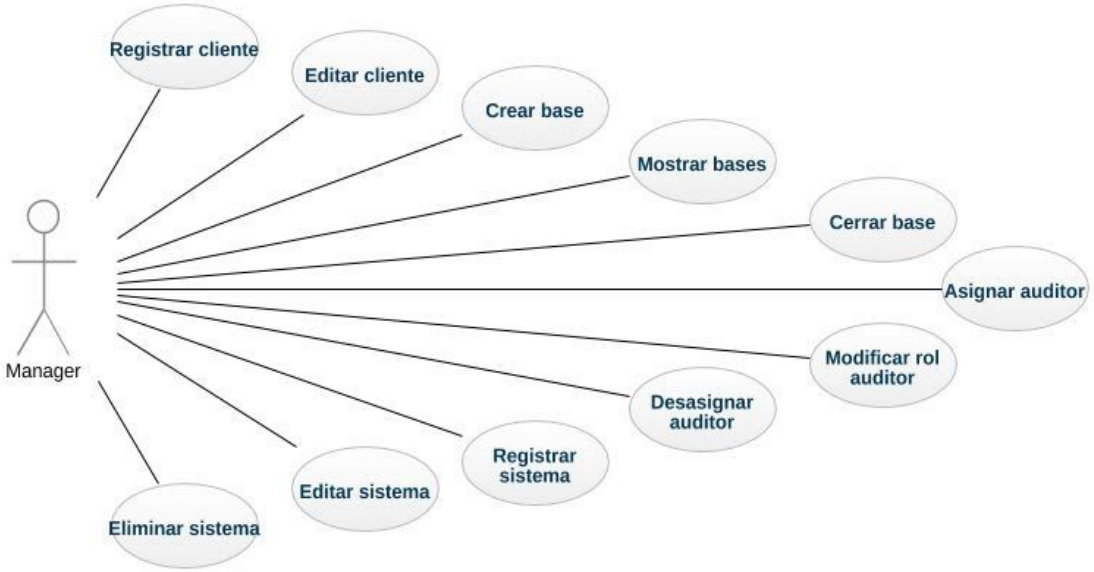


Figura 29. Diagrama de casos de uso para la gestión de bases

6.- Gestión de clientes

- **CU-19 - Registrar cliente:**
  - El usuario selecciona la opción “Clientes” y luego selecciona la opción “Nuevo cliente”.
  - El sistema abre una nueva ventana en la que el usuario introduce el nombre del cliente y opcionalmente una descripción.
  - Una vez introducidos el usuario selecciona la opción “Guardar”.
  - En caso de que no exista un cliente con el mismo nombre el sistema crea y almacena el cliente.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-20 - Editar cliente:**
  - El usuario selecciona la opción “Clientes” y el sistema le muestra una lista con todos los clientes registrados.
  - El usuario filtra por nombre del cliente y selecciona el cliente a editar.
  - El sistema abre una ventana con la información del cliente y el usuario modifica el nombre y/o la descripción.
  - Una vez terminado selecciona la opción “Guardar”.
  - En el caso de que no exista un cliente con el mismo nombre el sistema actualiza la información del cliente.
  - En el caso contrario el sistema muestra un mensaje informando del error.

7.- Administración de bases

- **CU-21 - Crear base:**
  - El usuario selecciona la opción “Nueva base” y el sistema abre una ventana con el listado de clientes registrados en el sistema.
  - El usuario selecciona el cliente para el que se creará la base correspondiente a su auditoría y luego selecciona el año fiscal de esa auditoría.
  - En el caso de que no exista una base para el mismo cliente y año fiscal el sistema crea una base para ese cliente y año fiscal que tiene como estado “Activa” y al propio usuario asignado con el rol *Team Leader*.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-22 - Mostrar bases:**
  - El usuario selecciona la opción “Todas las bases” y el sistema abre una ventana en la que muestra un listado con todas las bases registradas en el sistema (se muestra cliente y período fiscal), el nombre del usuario creador de cada una de estas bases y el estado en el que se encuentra la base.

- **CU-23 - Cerrar base:**
  - El usuario selecciona la opción “Mis bases” y el sistema abre una ventana en la que se listan todas las bases del sistema creadas por el usuario.
  - El usuario selecciona una de las bases listadas y luego selecciona la opción “Cerrar base”.
  - El sistema abre una ventana con un mensaje alertando de que el cierre de la base es definitivo y no se puede deshacer y el usuario selecciona la opción “Cerrar”.
  - En caso de que no exista ningún EGA en la base en estado distinto a *Cerrado* el sistema cambia el estado de la base a *Cerrado.*
  - En caso contrario el sistema muestra un mensaje informando del error.

8.- Gestión de auditores

El usuario accede a los siguientes casos de uso al abrir una base creada por él.

- **CU-24 - Asignar auditor:**
  - El usuario selecciona la opción “Administración” > “Auditores asignados” en el menú de la base y luego selecciona la opción “Asignar auditor a la base”.
  - El sistema abre una ventana con el listado de todos los auditores del departamento de SPA, en el que el usuario puede filtrar por nombre y/o categoría en el departamento.
  - El usuario selecciona un auditor de este listado y el sistema abre una ventana en la que el usuario selecciona el rol con el que se añadirá el auditor a la base.
  - El usuario selecciona la opción “Asignar”.
  - El sistema asigna el auditor a la base con el rol seleccionado.

- **CU-25 - Modificar rol auditor:**
  - El usuario selecciona la opción “Administración” > “Auditores asignados” y el sistema le muestra un listado con los auditores asignados a la base y su rol en ella.
  - El usuario selecciona el auditor a modificar y el sistema abre una ventana en la que el usuario selecciona el nuevo rol del auditor.
  - El usuario selecciona la opción “Guardar”.
  - El sistema actualiza asigna el usuario a la base con el nuevo rol y elimina el anterior.

- **CU-26 - Desasignar auditor:**
  - El usuario selecciona la opción “Administración” > “Auditores asignados” y el sistema le muestra un listado con los auditores asignados a la base y su rol en ella.
  - El usuario selecciona el auditor a eliminar y luego selecciona la opción “Eliminar de la base”.
  - El sistema abre una ventana en la que pide la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - El sistema elimina al auditor de la base y asigna todos los EGAs que tenía asignados al auditor que venía a continuación en la cadena de asignación del EGA.

9.- Gestión de sistemas de clientes

El usuario accede a los siguientes casos de uso seleccionando la opción “Clientes” en el menú principal y luego seleccionando un cliente concreto del listado de clientes registrados en el sistema.

- **CU-27 - Registrar sistema:**
  - El usuario selecciona la opción “Administración” > “Sistemas cliente” y luego selecciona la opción “Nuevo sistema”.
  - El sistema abre una ventana en la que el usuario introduce el nombre del sistema, introduce una descripción y, opcionalmente, selecciona el sistema estándar, al que corresponde el sistema concreto, del listado de sistemas estándares registrados.
  - El usuario selecciona la opción “Guardar”.
  - En el caso de que no exista ningún sistema con el mismo nombre para este cliente concreto, el sistema crea y almacena el nuevo sistema.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-28 - Editar sistema:**
  - El usuario selecciona la opción “Administración” > “Sistemas cliente” y el sistema le muestra un listado con todos los sistemas del cliente concreto.
  - El usuario selecciona un sistema de este listado y luego selecciona la opción “Editar”.
  - El sistema abre una ventana con la información de ese sistema concreto.
  - El usuario puede modificar el nombre, la descripción, el alcance y/o seleccionar el sistema estándar del listado de sistemas estándares registrados que le corresponda.
  - El usuario selecciona la opción “Guardar”.
  - En el caso de que no exista ningún sistema con el mismo nombre para ese cliente concreto el sistema guarda las modificaciones realizadas.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-29 - Eliminar sistema:**
  - El usuario selecciona la opción “Administración” > “Sistemas cliente” y el sistema le muestra un listado con todos los sistemas del cliente concreto.
  - El usuario selecciona un sistema de este listado y luego selecciona la opción “Eliminar sistema”.
  - El sistema abre una ventana en la que se pide la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - En caso de que el sistema del cliente no tenga ningún control asociado en ninguna de las bases del cliente el sistema lo elimina.
  - En el caso contrario el sistema muestra un mensaje informando del error.

Auditor - Team Leader

Los casos de uso del actor correspondiente al auditor con rol *Team Leader* en una base dan respuesta a las dos funcionalidades específicas de ese rol en la base: *10.- Seleccionar áreas e ITGCs* y *11.- Cerrar EGAs.*

Todos los casos de uso parten de la precondición de que el usuario ha seleccionado una base en estado *Activa* y la ha abierto.

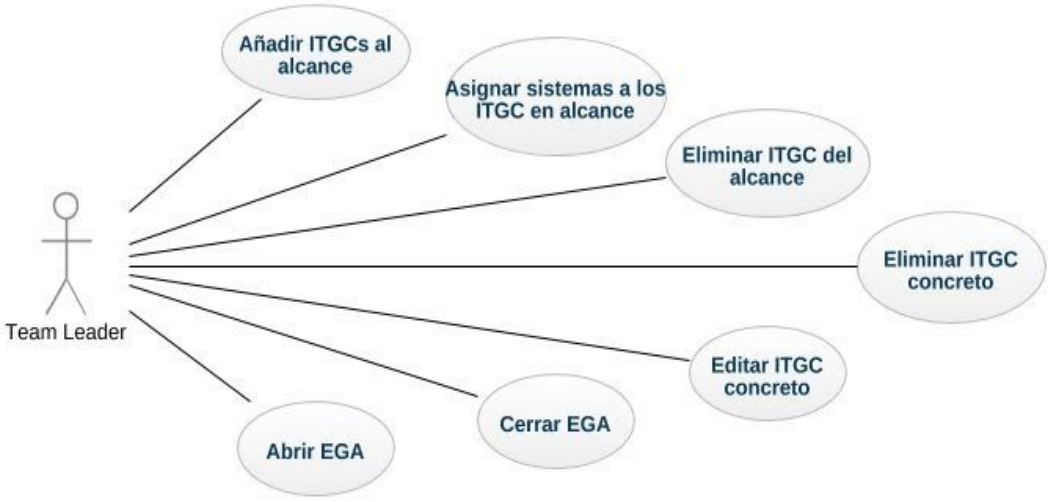


Figura 30. Diagrama de casos de uso las funcionalidades del *Team Leader*

10.- Gestionar el alcance de la base

- **CU-30 - Añadir ITGCs al alcance:**
  - El usuario selecciona la opción “Administración” > “Alcance” y luego selecciona la opción “Añadir controles”.
  - El sistema abre una ventana en la que se muestran todos los ITGCs estándares, agrupados por área de aplicación, que no se encuentran ya en el alcance de la base.
  - El usuario selecciona los ITGCS estándares deseados de cada área y una vez terminado selecciona la opción “Finalizar”.
  - El sistema crea un ITGC concreto de tipo transversal con el mismo título, descripción, tipo y área que su ITGCs estándar para todos los controles seleccionados.

- **CU-31 - Asignar sistemas a los ITGC en alcance:**
  - El usuario selecciona la opción “Administración” > “Alcance” y luego selecciona la opción “Sistemas por ITGC”.
  - El sistema abre una ventana con el listado de los ITGCs estándares en el alcance de la base y que son de tipo “Específicos de sistema”, para cada ITGC se muestra un listado con los sistemas en alcance del cliente.
  - El usuario marca de este listado los sistemas a los que aplica cada control.
  - El usuario selecciona la opción “Finalizar” una vez seleccionados.
  - El sistema crea un ITGC específico de sistema, para cada sistema seleccionado, asociado al ITGC estándar.
  - Si el ITGC ya estaba presente en la base para ese sistema concreto no se realizará ninguna modificación.

- **CU-32 - Eliminar ITGC del alcance:**
  - El usuario selecciona la opción “Administración” > “Alcance” y el sistema abre una nueva vista con el listado de todos los ITGCs estándares presentes en el alcance de la base.
  - El usuario selecciona un ITGC de este listado y luego selecciona la opción “Eliminar del alcance”.
  - El sistema abre una ventana donde solicita la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - El sistema elimina todos los ITGCs relacionados con ese ITGC estándar en el caso de que no tengan ningún EGA asociado.
  - En caso contrario el sistema muestra un mensaje informando del error.

- **CU-33 - Eliminar ITGC concreto:**
  - El usuario selecciona un ITGC concreto de la base y luego selecciona la opción “Eliminar ITGC”.
  - El sistema abre una ventana donde solicita la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - En el caso de que el ITGC concreto seleccionado no tenga ningún EGA asociado el sistema lo elimina de la base.
  - En el caso contrario el sistema muestra un mensaje informando del error.
  - Si no existe otro ITGC en la base relacionado con el ITGC estándar del ITGC eliminado, el sistema elimina el ITGC estándar del alcance.

- **CU-34 - Editar ITGC concreto:**
  - El usuario selecciona un ITGC concreto de la base y luego selecciona la opción “Editar ITGC”.
  - El sistema abre una ventana con la información de ese ITGC concreto y el usuario puede modificar su descripción.
  - El usuario selecciona la opción “Guardar”.
  - El sistema guarda la modificación realizada.

11.- Cerrar EGAs

- **CU-35 - Cerrar EGA:**
    - El usuario selecciona un EGA de la base.
    - Sí su estado es *Revisado* y está asignado a él:
        - El sistema muestra la opción “Cerrar EGA” y el usuario la selecciona.
        - El sistema abre una ventana donde solicita la confirmación del usuario y éste selecciona la opción “Cerrar”.
        - El sistema modifica el estado del EGA y lo actualiza a *Cerrado*, de forma que no admite ninguna modificación por parte de ningún auditor y solo se puede consultar en modo lectura.
    - En caso contrario, el sistema no muestra esta opción.

- **CU-36 - Abrir EGA:**
    - El usuario selecciona un EGA de la base.
    - Si su estado es *Cerrado*:
        - El sistema muestra la opción “Abrir EGA” y el usuario la selecciona.
        - El sistema abre una ventana donde solicita la confirmación del usuario y éste selecciona la opción “Abrir”.
        - El sistema modifica el estado del EGA y lo actualiza a *Revisado,* de forma que puede volver a ser modificado.

Auditor - Reviewer

Los casos de uso del actor correspondiente al auditor con rol *Reviewer* en una base dan respuesta a las funcionalidades específicas de este rol, relacionadas principalmente con la gestión de los requerimientos de información y la revisión de los EGAs.

Las funcionalidades relacionadas con los requerimientos de información se han analizado en el grupo denominado *Gestión de clientes y requerimientos,* formado por las siguientes funcionalidades: *12.- Administración de responsables de IT de los clientes, 13.- Gestión de requerimientos de información, 14.- Asignación de requerimientos a los clientes* y *15.- Gestión de recordatorios,* la funcionalidad *16.- Revisión de EGAs* se ha analizado de forma individual.

Todos los casos de uso parten de la precondición de que el usuario ha seleccionado una base en estado *Activa* y la ha abierto.

Gestión de responsables IT y requerimientos

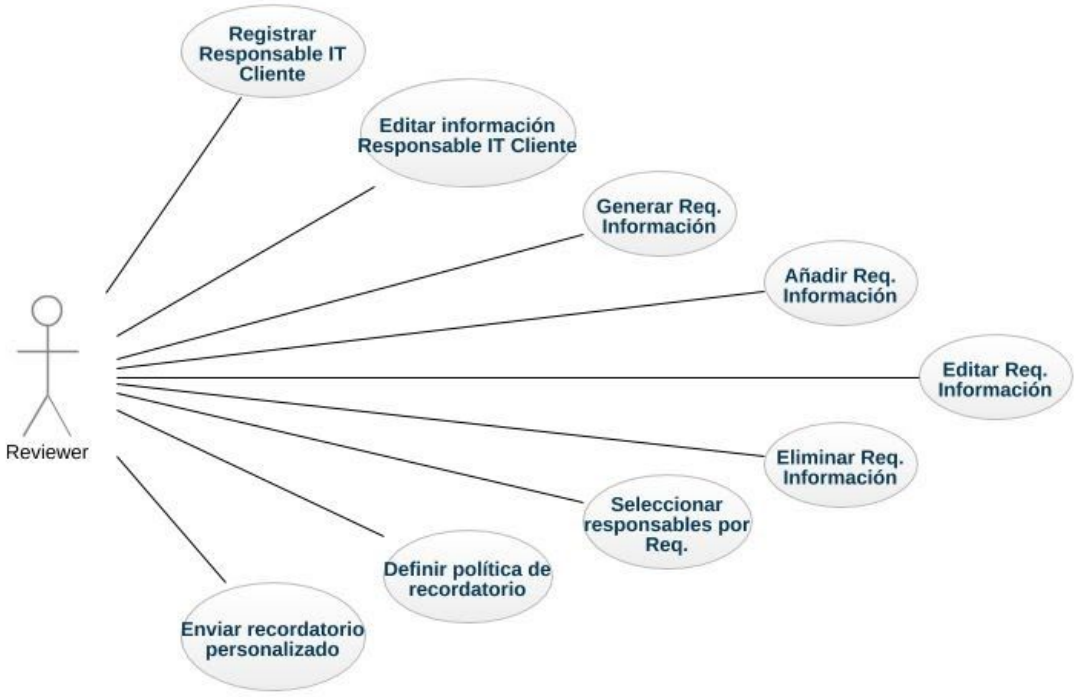


Figura 31. Diagrama de casos de uso para la gestión de responsables IT y requerimientos

12.- Administración de responsables de IT de los clientes

- **CU-37 - Registrar Responsable IT Cliente:**
  - El usuario selecciona la opción “Administración” > “Responsables IT Cliente” y luego selecciona la opción “Registrar”.
  - El sistema abre una nueva ventana en la que el usuario introduce el email, el nombre, y, opcionalmente, la descripción del rol y posición del responsable IT a registrar.
  - El usuario selecciona la opción “Registrar” una vez introducidos los campos.
  - En el caso de que no exista ningún contacto de IT con el mismo email, el sistema crea el usuario correspondiente al responsable de IT, le genera una contraseña aleatoria y le envía automáticamente link para acceder al sistema y cambiarla.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-38 - Editar información Responsable IT Cliente:**
  - El usuario selecciona la opción “Administración” > “Responsables IT Cliente” y el sistema abre una ventana en la que se listan todos los responsables de IT registrados para ese cliente.
  - El usuario selecciona un responsable del listado y luego selecciona la opción “Editar”.
  - El sistema abre una nueva ventana en la que se muestra el nombre, correo y descripción del contacto seleccionado.
  - El usuario modifica el nombre y/o la descripción.
  - El usuario selecciona la opción “Guardar”.
  - El sistema actualiza los cambios realizados.

13.- Gestión de requerimientos de información

- **CU-39 - Generar Req. Información:**
  - El usuario selecciona la opción “Requerimientos de Información” y luego selecciona la opción “Generar Req. Información estándares”.
  - El sistema abre una nueva ventana solicitando la confirmación del usuario, el usuario selecciona la opción “Aceptar” y el sistema genera automáticamente los requerimientos de información para la base.

  Estos requerimientos se generan en función de los ITGCs presentes en la base y sus sistemas asociados, de forma que si en el sistema existe un Req. de información Estándar asignado a un ITGC Estándar y este se encuentra representado por un ITGC de la base, se creará un Req. de información con estado *Pendiente* asociado a ese ITGC de la base con el mismo nombre, descripción, área y grupo que el que tenga el Req. de información Estándar, el sistema también asignará un identificador en función del área, grupo y del orden en el que sean creados. Del mismo modo, si existe un Req. de información Estándar de tipo *Específico de sistema* asignado a un ITGC Estándar y a un Sistema Estándar y en la base existe ese ITGC asociado al mismo sistema del cliente, también se creará un Req. de información con estado *Pendiente* asociado a ese ITGC de la base.

- **CU-40 - Añadir Req. Información:**
  - El usuario selecciona la opción “Requerimientos de información” y luego selecciona la opción “Añadir Requerimiento”.
  - El sistema abre una nueva ventana en la que el usuario introduce el identificador del requerimiento, su nombre, descripción, la información requerida y selecciona el área de aplicación del requerimiento y el grupo al que pertenece (el sistema sólo mostrará aquellos grupos con la misma área de aplicación que la seleccionada), por último el sistema muestra el listado de ITGCs de la base por la área de aplicación seleccionada y el usuario selecciona aquellos que se relacionen con el requerimiento.
  - El usuario selecciona la opción “Guardar”.
  - En el caso de que no exista otro requerimiento con el mismo identificador, el sistema crea y guarda en la base el nuevo requerimiento.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-41 - Editar Req. Información:**
  - El usuario selecciona la opción “Requerimientos de información” y el sistema abre una nueva vista con el listado de todos los Req. de información de la base.
  - El usuario selecciona el requerimiento deseado y luego selecciona la opción “Editar”.
  - El sistema abre una nueva ventana en la que muestra la información del requerimiento y el usuario modifica el identificador, el nombre, la descripción, la información requerida, selecciona el grupo del requerimiento y/o selecciona los controles que se relacionen con el requerimiento.
  - El usuario selecciona la opción “Guardar”.
  - En el caso de que no exista otro requerimiento con el mismo identificador el sistema actualiza la información del requerimiento seleccionado.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-42 - Eliminar Req. Información:**
  - El usuario selecciona la opción “Requerimientos de Información” y el sistema abre una ventana con el listado de todos los Req. de información de la base.
  - El usuario selecciona el requerimiento deseado y luego selecciona la opción “Eliminar”.
  - El sistema abre una ventana donde solicita la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - En el caso de que el requerimiento no tenga ninguna evidencia el sistema lo elimina de la base.
  - En el caso contrario el sistema muestra un mensaje informando del error.

14.- Asignación de requerimientos a sus responsables

- **CU-43 - Seleccionar responsable por área:**
  - El usuario selecciona la opción “Requerimientos de Información” y el sistema abre una ventana en la que el usuario selecciona la opción “Seleccionar responsables”.
  - El sistema abre una ventana en la que solicita al usuario elegir el tipo de selección a realizar y el usuario elige el tipo “Selección por área”.
  - El sistema abre una ventana en la que se muestra un listado con las áreas de aplicación presentes para los requerimientos de información de esa base.
  - El usuario selecciona una área del listado y el sistema despliega un listado con los responsables de IT registrados para el cliente de la base.
  - El usuario selecciona y/o deselecciona los responsables que desee del listado mostrado.
  - El usuario selecciona la opción “Actualizar”.
  - El sistema elimina a los anteriores responsables asignados al requerimiento.
  - El sistema asigna los responsables seleccionados a todos los requerimientos de información del área seleccionada.

- **CU-44 - Seleccionar responsable por requerimiento:**
  - El usuario selecciona la opción “Requerimientos de Información” y el sistema abre una ventana en la que el usuario selecciona la opción “Seleccionar responsables”.
  - El sistema abre una ventana en la se solicita al usuario elegir el tipo de selección y el usuario elige el tipo “Selección por requerimiento”.
  - El sistema abre una ventana en la que se muestra un listado con todos los requerimientos de información de la base agrupados por su área de aplicación.
  - El usuario selecciona un requerimiento del listado y el sistema despliega un listado con los responsables de IT registrados para el cliente de la base. Si uno de los responsables ya tiene ese requerimiento, aparecerá como seleccionado en el listado.
  - El usuario selecciona y/o deselecciona los responsables que desee de ese listado.
  - El usuario selecciona la opción “Actualizar”.
  - El sistema elimina a los anteriores responsables asignados al requerimiento.
  - El sistema asigna los responsables seleccionados al requerimiento de información seleccionado.

15.- Gestión de recordatorios

- **CU-45 - Definir política de recordatorios:**
  - El usuario selecciona la opción “Requerimientos de Información” y luego selecciona la opción “Recordatorios”.
  - El sistema abre una nueva ventana en la que se muestra la política de recordatorios y el usuario selecciona la opción “Editar”.
  - El usuario define el intervalo de tiempo en días entre recordatorios, selecciona si estos recordatorios se enviarán siempre o sólo si no se ha actualizado ningún requerimiento desde el último recordatorio (Opciones *Siempre* o *Si no actividad* respectivamente) y opcionalmente modifica el texto que recibe al cliente.
  - El usuario selecciona la opción “Guardar”.
  - El sistema guarda la política de recordatorios para esa base.

- **CU-46 - Enviar recordatorio:**
  - El usuario selecciona la opción “Requerimientos de Información” y luego selecciona la opción “Recordatorios”.
  - El sistema abre una nueva ventana en la que se muestra la política de recordatorios.
  - El usuario selecciona la opción “Enviar recordatorio manual” y el sistema abre una nueva ventana en la que aparece el texto del recordatorio a enviar.
  - El usuario modifica este texto y opcionalmente adjunta archivos al mensaje.
  - El usuario selecciona la opción “Siguiente” y el sistema abre una ventana con el listado de responsables de IT del cliente registrados.
  - El usuario selecciona a los contactos a los que quiere enviar el recordatorio del listado y una vez finalizado selecciona la opción “Enviar”.
  - El sistema envía el recordatorio al correo de los responsables seleccionados.

Revisión de EGAs

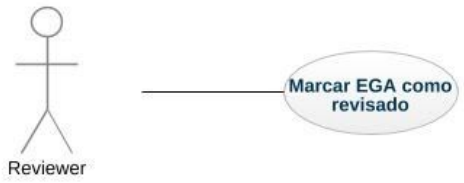


Figura 32. Diagrama de casos de uso para la revisión de EGAs

16.- Revisión de EGAs

- **CU-47 - Marcar EGA como revisado:**
  - El usuario selecciona un EGA de la base.
  - Si su estado es *Preparado* y se encuentra asignado a él:
    - El sistema muestra la opción “Marcar EGA como revisado”.
    - El usuario selecciona la opción “Marcar EGA como revisado”.
    - El sistema modifica el estado del EGA, que pasa a *Revisado,* y asigna el EGA al siguiente auditor de la jerarquía.

Auditor - Team Member

Los casos de uso del Auditor con rol *Team Member* en una base dan respuesta a las funcionalidades generales de todos los auditores, debido que los demás roles también tendrán acceso a estas funcionalidades, y están relacionadas principalmente con el acceso al sistema y sus bases, la gestión de evidencias, la creación/edición de EGAs y otras funcionalidades secundarias de la base. De cara a su análisis, se han agrupado las funcionalidades a las que dan respuesta en función de la relación entre ellas, de forma que se han obtenido 4 grupos formados por distintas funcionalidades que se listan a continuación:

1. **Acceso al sistema y a sus bases:** Agrupa a las funcionalidades *17.- Acceso al sistema* y *18.- Acceso a las bases.*

2. **Gestión de evidencias y ficheros tratados*:*** Agrupa a las funcionalidades *19.- Clasificación de evidencias, 20.- Administración manual de evidencias, 21.- Administración de ficheros tratados* y *22.- Recibir notificaciones.*

3. **Gestión de EGAs:** Agrupa las funcionalidades *23.- Administración de EGAs, 24.- Administración de la cadena de asignación, 25.- Edición de EGAs, 26.- Bloqueo de EGAs* y *27.- Preparar EGAs.*

4. **Otras funcionalidades:** Agrupa las funcionalidades *28.- Filtrar por condiciones, 29.- Consultar el dashboard, 30.- Administración de Coaching Notes* y *31.- Administración de Control Deficiencies.*

Acceso al sistema y a sus bases

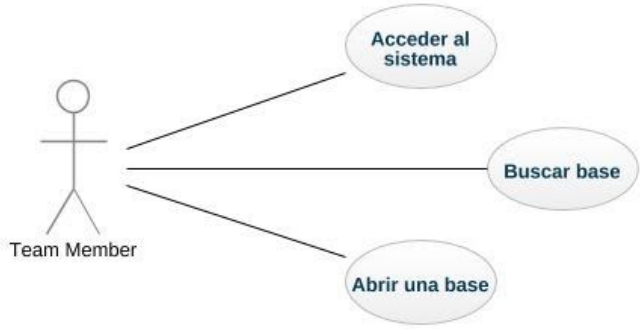


Figura 33. Diagrama de casos de uso para el acceso al sistema y a sus bases

17.- Acceso al sistema

- **CU-48 - Acceder al sistema:**
  - El usuario selecciona el icono del sistema en su dispositivo y se abre una ventana con el menú inicial del sistema.
  - El sistema identifica automáticamente al usuario con las credenciales del SD almacenadas en su dispositivo.

18.- Acceso a las bases

- **CU-49 - Buscar base:**
  - El usuario selecciona la opción “Mis bases” y el sistema abre una ventana con el listado de todas las bases asignadas al usuario.
  - El usuario filtra las bases seleccionando el campo “Cliente”, del listado de clientes registrados, el campo “Período fiscal” del listado de períodos, y/o el campo “Estado de la base”, que será *Activa* o *Cerrada*.
  - El sistema muestra las bases que cumplan con los filtros especificados por el usuario.

- **CU-50 - Abrir una base:**
  - El usuario selecciona la opción “Mis bases” y el sistema abre una ventana con el listado de todas las bases asignadas al usuario.
  - El usuario selecciona una de estas bases y selecciona la opción “Abrir”.
  - El sistema abre una nueva ventana con el menú asociado a esta.

Gestión de evidencias y ficheros tratados

Todos los casos de uso de esta funcionalidad parten de la precondición de que el usuario ha abierto una base activa asociada a él.

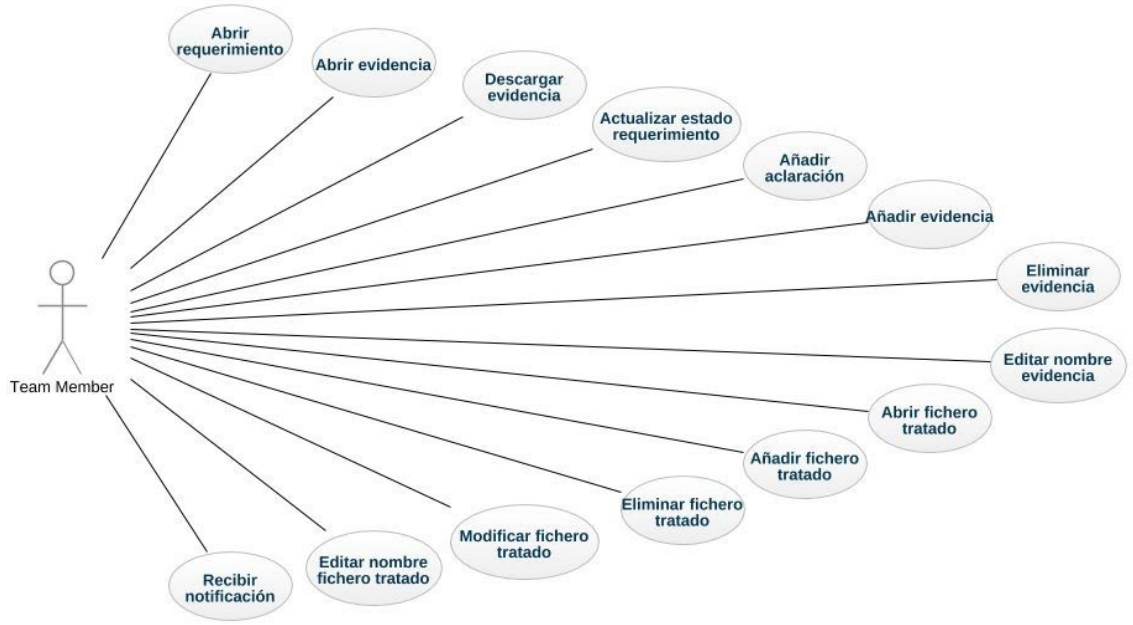


Figura 34. Diagrama de casos de uso para la gestión de evidencias y ficheros tratados

19.- Clasificación de evidencias

- **CU-51 - Abrir requerimiento:**

- El usuario selecciona la opción “Requerimientos de información” en el menú principal de la base y el sistema abre una nueva ventana con el listado de todos los requerimientos de información agrupados por su área de aplicación.
- El usuario selecciona el área de aplicación deseada y el sistema despliega los requerimientos de información de esa área.
- El usuario selecciona uno de estos requerimientos y selecciona la opción “Abrir”.
- El sistema despliega un listado con la información del requerimiento y el listado de todas las evidencias subidas para este, para cada evidencia muestra el usuario que la ha subido y los EGAs que enlazan a esta.

- **CU-52 - Abrir evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia y luego selecciona la opción “Abrir”.
  - El sistema abre el fichero que contiene la evidencia seleccionada en modo lectura, de forma que el auditor puede consultar la información pero no puede realizar ninguna modificación.

- **CU-53 - Descargar evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia y luego selecciona la opción “Descargar”.
  - El sistema descarga el fichero que contiene la evidencia seleccionada en el dispositivo del auditor.

- **CU-54 - Actualizar estado requerimiento:**
  - El usuario abre un requerimiento y luego selecciona la opción “Actualizar”.
  - El sistema abre una ventana en la que se muestra la información del requerimiento y su estado actual.
  - El usuario selecciona el nuevo estado del requerimiento, que puede ser “Pendiente” o “Recibido”, y luego selecciona la opción “Guardar”.
  - El sistema guarda el nuevo estado del requerimiento.

- **CU-55 - Añadir aclaración:**
  - El usuario abre un requerimiento y luego selecciona la opción “Actualizar”.
  - El sistema abre una ventana en la que se muestra la información del requerimiento y su estado actual.
  - El usuario selecciona la opción “Añadir aclaración” y el sistema muestra un campo de texto titulado “Aclaración”
  - El usuario introduce la aclaración a mostrar en ese campo y luego selecciona la opción “Guardar”.
  - El sistema actualiza el campo “Aclaración” introducido.

20.- Administración manual de evidencias

- **CU-56 - Añadir evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona la opción “Nueva evidencia” y el sistema abre una nueva ventana donde usuario introduce el nombre de la evidencia y selecciona de su dispositivo el fichero a subir que la contiene.
  - El usuario selecciona la opción “Guardar”.
  - En el caso que no exista una evidencia con el mismo nombre para ese requerimiento, el sistema almacena la nueva evidencia subida.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-57 - Eliminar evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia del listado y luego selecciona la opción “Eliminar”.
  - El sistema muestra un mensaje solicitando la confirmación del usuario.
  - El usuario selecciona la opción “Eliminar”.
  - El sistema elimina la evidencia seleccionada, aunque guarda una copia interna por si fuera necesario restaurarla en el futuro.

- **CU-58 - Editar nombre evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia del listado y luego selecciona la opción “Editar”.
  - El sistema permite al usuario modificar el campo con el nombre de la evidencia y el usuario introduce el nuevo nombre deseado.
  - El usuario selecciona la opción “Guardar”.
  - En el caso que no exista una evidencia con el mismo nombre para ese requerimiento el sistema actualiza el nombre de la evidencia.
  - En el caso contrario el sistema muestra un mensaje informando del error.

21.- Administración de ficheros tratados

- **CU-59 - Abrir fichero tratado:**
  - El usuario selecciona la opción “Ficheros” en el menú principal de la base y el sistema abre una nueva ventana con el listado de todos los ficheros tratados de la base agrupados por su área de aplicación.
  - El usuario puede filtrar los resultados introduciendo el campo “Nombre”, seleccionando el área de aplicación del fichero y luego seleccionando la opción “Filtrar”
  - El sistema filtra los ficheros en función de los parámetros introducidos.
  - El usuario selecciona el fichero tratado y luego selecciona la opción “Abrir”.
  - El sistema abre ese fichero en el dispositivo del auditor.

- **CU-60 - Añadir fichero tratado:**
  - El usuario selecciona la opción “Ficheros” en el menú principal de la base y luego selecciona la opción “Nuevo fichero tratado”.
  - El sistema abre una nueva ventana en la que el usuario selecciona de su dispositivo el fichero tratado y una vez subido selecciona la opción “Guardar”.
  - En el caso de que no exista ningún fichero tratado en la base con el mismo nombre que el fichero subido el sistema crea y guarda el fichero tratado en la base. (El nombre del fichero tratado corresponderá al nombre que tenía ese fichero en el dispositivo del auditor en el momento que se subió).
  - En caso contrario el sistema muestra un mensaje informando del error.

- **CU-61 - Eliminar fichero tratado:**
  - El usuario selecciona la opción “Ficheros” en el menú principal de la base y el sistema abre una nueva ventana con el listado de todos los ficheros tratados de la base, agrupados por su área de aplicación.
  - El usuario selecciona un fichero tratado del listado y luego selecciona la opción “Eliminar”.
  - El sistema muestra un mensaje solicitando la confirmación del usuario y el usuario selecciona la opción “Eliminar”.
  - El sistema elimina el fichero tratado.

- **CU-62 - Modificar fichero tratado:**
  - El usuario selecciona la opción “Ficheros” en el menú principal de la base y el sistema abre una nueva ventana con el listado de todos los ficheros tratados de la base, agrupados por su área de aplicación.
  - El usuario selecciona un fichero tratado de la base y luego selecciona la opción “Abrir”.
  - El sistema abre ese fichero con el programa del dispositivo del auditor adecuado.
  - El usuario modifica el fichero y una vez finalizado selecciona la opción “Guardar”.
  - El sistema guarda en la base los cambios realizados en el fichero.

- **CU-63 - Editar nombre fichero tratado:**
  - El usuario selecciona la opción “Ficheros” en el menú principal de la base y el sistema abre una nueva ventana con el listado de todos los ficheros tratados de la base, agrupados por su área de aplicación.
  - El usuario selecciona un fichero tratado del listado y luego selecciona la opción “Editar Nombre”.
  - El sistema permite al usuario modificar el campo con el nombre del fichero tratado y el usuario introduce el nuevo nombre deseado.
  - El usuario selecciona la opción “Guardar”.
  - En el caso que no exista un fichero tratado con el mismo nombre en la base el sistema actualiza el fichero con el nuevo nombre.
  - En el caso contrario el sistema muestra un mensaje informando del error.

22.- Recibir notificaciones

- **CU-64 - Recibir notificación:**
  - El responsable de IT del cliente asignado a un requerimiento sube una o varias evidencias para uno o varios requerimientos.
  - El sistema envía un *mail* automáticamente notificando al usuario que se han subido evidencias para cierta base que tiene asignada.

Gestión de EGAs

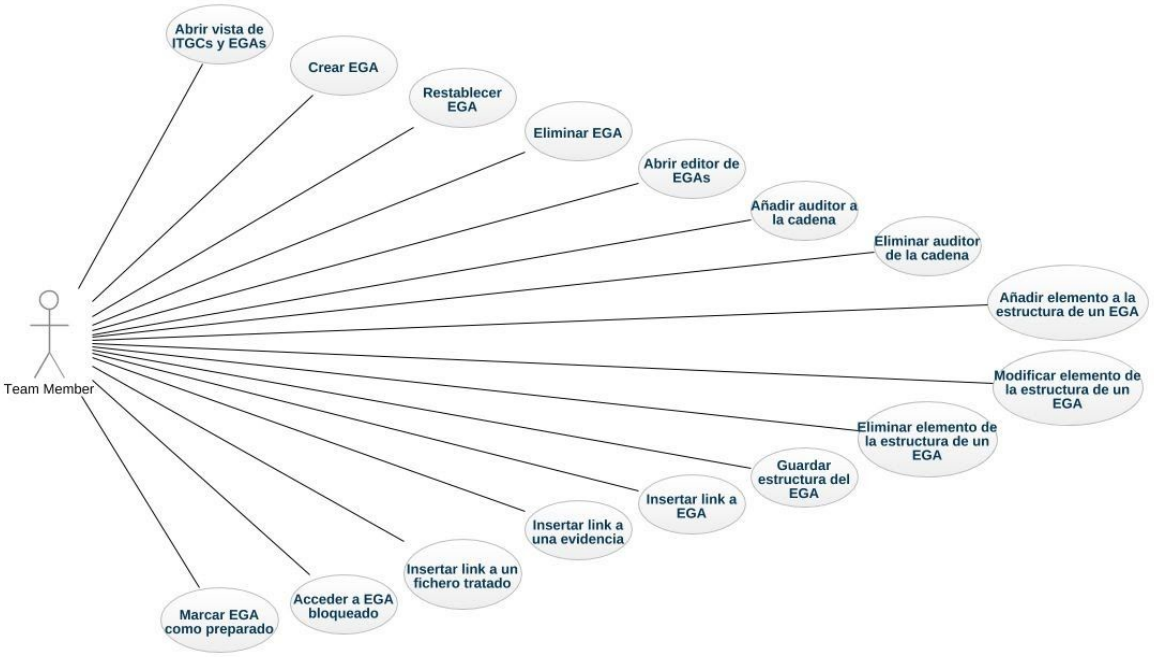


Figura 35. Diagrama de casos de uso para la gestión de EGAs

23.- Administración de EGAs

- **CU-65 - Abrir vista de ITGCs y EGAs:**
  - El usuario selecciona la opción “ITGCs” en el menú principal de la base y el sistema le muestra un listado con todos los ITGCs concretos presentes en la base agrupados por área de aplicación.
  - El usuario selecciona una área de aplicación.
  - El sistema despliega los ITGCs concretos de esa área (en caso de que existan varios ITGCs que representen el mismo ITGC genérico, estos se agruparan en el genérico y al seleccionarlo se desplegaran los concretos) y el EGA asociado a cada ITGC concreto en caso de que exista.
  - Para cada ITGC concreto muestra su estado y si tiene algun *Control Deficiency* asociado, para cada EGA muestra su estado y el auditor al que está asignado.

- **CU-66 - Crear EGA:**
  - El usuario selecciona un ITGC concreto de la base y luego selecciona la opción “Crear EGA”.
  - El sistema abre una nueva ventana en la que muestra un listado con el nombre de las plantillas registradas disponibles para ese ITGC, las plantillas se encuentran agrupadas en función de si son genéricas o específicas de sistema (esta segunda opción solo estará disponible si el ITGC seleccionado es específico de sistema).
  - El usuario puede seleccionar una plantilla y luego seleccionar la opción “Previsualizar”, de forma que el sistema abre la plantilla en modo lectura para que el usuario vea su estructura.
  - El usuario selecciona la plantilla deseada y luego selecciona la opción “Crear”.
  - El sistema crea un EGA en la base asociado al ITGC concreto y con la misma estructura que la de la plantilla seleccionada, asigna el EGA al usuario creador y en la cadena de asignación añade al usuario creador, a un auditor de la base con rol “Reviewer” como su revisor y por encima al “Team Leader” de la base.
  - En caso de que ya exista un EGA para ese ITGC, el sistema muestra un mensaje informando del error al seleccionar la opción “Crear EGA”.

- **CU-67 - Restablecer EGA:**
  - El usuario selecciona un EGA de la base, luego selecciona la opción “Restablecer”.
  - El sistema muestra un mensaje donde alerta de que la acción eliminará toda la información contenida en la estructura actual y solicita la confirmación de este.
  - El usuario selecciona la opción “Restablecer”.
  - El sistema muestra un listado con el nombre de las plantillas registradas disponibles para el ITGC asociado al EGA.
  - El usuario selecciona la plantilla deseada y luego selecciona la opción “Crear”.
  - El sistema actualiza la estructura del EGA con la estructura de la plantilla seleccionada, eliminando por completo toda la información contenida en la estructura anterior.

- **CU-68 - Eliminar EGA:**
  - El usuario selecciona un EGA de la base y luego selecciona la opción “Eliminar”.
  - El sistema muestra un mensaje solicitando la confirmación del usuario y éste selecciona la opción “Eliminar”.
  - El sistema elimina el EGA de la base.

- **CU-69 - Abrir EGA:**
  - El usuario selecciona un EGA de la base y luego selecciona la opción “Abrir”.
  - El sistema abre una nueva ventana en la que el usuario puede editar la estructura del EGA.
  - El sistema pone el EGA en modo “Bloqueado”, de forma que si otro auditor abre ese mismo EGA en paralelo no podrá realizar ninguna modificación.
  - El usuario cierra la ventana en la que editaba la estructura del EGA.
  - El sistema desbloquea el EGA poniendolo en modo “Abierto”

24.- Administración de la cadena de asignación

La cadena de asignación de un EGA debe cumplir siempre las siguientes condiciones:

- Un auditor con el rol *Team Member* no puede revisar el trabajo de otro auditor.
- Un auditor con el rol *Reviewer* puede revisar el trabajo de un auditor *Team Member*.
- Debe haber siempre un auditor con el rol *Team Leader* al final de la cadena como revisor final del trabajo realizado por los otros auditores.

- **CU-70 - Añadir auditor a la cadena:**
  - El usuario selecciona un EGA de la base y luego selecciona la opción “Cadena de asignación”.
  - El sistema abre una nueva ventana en la que aparecen los auditores que forman parte de la cadena de asignación de ese EGA ordenados en función de su posición en la cadena.
  - El usuario selecciona la opción “Añadir”.
  - El sistema muestra un listado con todos los auditores asignados a la base, donde el usuario puede filtrar por nombre del auditor.
  - El usuario selecciona el auditor a añadir y luego selecciona la posición de la cadena en la que se añadirá, de forma que el auditor en la base de la cadena será el encargado de preparar el EGA y los auditores situados por encima suyo revisarán su trabajo.
  - Una vez finalizado el usuario selecciona la opción “Guardar”.
  - En el caso de que la cadena modificada cumpla con las condiciones comentadas anteriormente, el sistema guarda la nueva cadena de asignación.
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-71 - Eliminar auditor de la cadena:**
  - El usuario selecciona un EGA de la base y luego selecciona la opción “Cadena de asignación”.
  - El sistema abre una nueva ventana en la que aparecen los auditores que forman parte de la cadena de asignación de ese EGA ordenados en función de su posición en la cadena.
  - El usuario selecciona uno de los auditores de la cadena y luego selecciona la opción “Eliminar”.
  - El sistema elimina al auditor de la cadena de asignación.
  - Una vez finalizado el usuario selecciona la opción “Guardar”.
  - En el caso de que la cadena modificada cumpla con las condiciones comentadas anteriormente, el sistema guarda la nueva cadena de asignación.
  - En el caso contrario el sistema muestra un mensaje informando del error.

25.- Edición de EGAs

- **CU-72 - Añadir elemento a la estructura de un EGA:**
  - El usuario selecciona el contenedor del elemento que va a añadir a la estructura, excepto si el elemento es una hoja, que en este caso se añadirá directamente.
  - El sistema muestra el listado de elementos a añadir.
  - El usuario elige el elemento deseado.
  - El sistema crea este elemento seleccionado dentro de su contenedor.

- **CU-73 - Modificar elemento de la estructura de un EGA:**
  - El usuario selecciona un elemento de la estructura.
  - El sistema muestra todas las opciones para la modificación del elemento.
  - El usuario selecciona la opción y modifica el elemento.

- **CU-74 - Eliminar elemento de la estructura de un EGA:**
  - El usuario selecciona un elemento de la estructura y luego selecciona la opción “Eliminar”.
  - El sistema elimina el elemento de la estructura.

- **CU-75 - Insertar link a otro EGA:**
  - El usuario selecciona el elemento que contendrá el link, que será siempre un elemento de tipo “Comentario” o “Tabla”, y luego selecciona la opción “Añadir link”.
  - El sistema abre una nueva ventana en la que lista los tipos de links que se pueden insertar y el usuario selecciona la opción “EGA”.
  - El sistema abre una nueva ventana en la que se listan todos los EGAs de la base agrupados por control y área de aplicación.
  - El usuario selecciona el EGA deseado y selecciona la opción “Insertar”.
  - El sistema añade el link a este EGA en el elemento contenedor.

- **CU-76 - Insertar link a evidencia:**
  - El usuario selecciona el elemento que contendrá el link, que será siempre un elemento de tipo “Apartado”, y luego selecciona la opción “Añadir link”.
  - El sistema abre una nueva ventana en la que lista los tipos de links que se pueden insertar y el usuario selecciona la opción “Evidencia”.
  - El sistema abre una nueva ventana en la que muestra las evidencias subidas por los requerimientos asociados al ITGC del EGA y agrupadas por estos requerimientos.
  - El usuario puede seleccionar la opción “Todas las evidencias” y el sistema muestra las evidencias subidas para todos los requerimientos de información del sistema agrupados por requerimiento y a su vez por área de aplicación.
  - El usuario selecciona una de las evidencias listadas y selecciona la opción “Insertar”.
  - El sistema añade el link a la evidencia dentro de un elemento titulado “Evidencias” ubicado dentro del apartado seleccionado como contenedor.

- **CU-77 - Insertar link a fichero tratado:**
  - El usuario selecciona el elemento que contendrá el link, que será siempre un elemento de tipo “Apartado”, y luego selecciona la opción “Añadir link”.
  - El sistema abre una nueva ventana en la que lista los tipos de links que se pueden insertar y el usuario selecciona la opción “Fichero tratado”.
  - El sistema abre una nueva ventana en la que se listan todos los ficheros tratados asociados a la área de aplicación del ITGC.
  - El usuario selecciona uno de los ficheros listados y selecciona “Insertar”.
  - El sistema añade el link a este fichero dentro de un elemento titulado “Ficheros tratados” ubicado dentro del apartado seleccionado como contenedor.

- **CU-78 - Guardar estructura del EGA:**
  - El usuario selecciona la opción “Guardar”.
  - El sistema guarda la estructura del EGA teniendo en cuenta todos los elementos que han sido añadidos, modificados y/o eliminados.

26.- Bloqueo de EGAs

- **CU-79 - Acceder a EGA bloqueado:**
  - El usuario selecciona un EGA de la base al que está accediendo otro usuario y por lo tanto se encuentra bloqueado.
  - El sistema le muestra un mensaje indicando que el EGA está siendo accedido y que sólo puede ser abierto en modo lectura.
  - El usuario selecciona la opción “Abrir” y el sistema abre la estructura del EGA en modo lectura, de forma que no se puede modificar ningún elemento.

27.- Preparar EGAs

- **CU-80 - Marcar EGA como preparado:**
  - El usuario selecciona un EGA de la base que esté asignado a él y luego selecciona la opción “Marcar como preparado”.
  - El sistema abre una nueva ventana en la que el usuario selecciona el estado del control entre las dos opciones que le ofrece el sistema, que son “Sin excepciones” o “Con excepciones”.
  - En caso de que el usuario seleccione la opción “Con excepciones” el sistema abre una ventana preguntando al usuario si quiere crear una *Control Deficiency.*
  - Si el usuario selecciona “Sí”:
    - El sistema se abre una nueva ventana para su creación.
  - Si el usuario selecciona “No”:
    - El sistema desplegará un campo de texto titulado “Justificación”, donde el usuario justificará por qué no se crea una deficiencia para ese control.
  - El usuario selecciona la opción “Preparar”.
  - El sistema cambia el estado del EGA de *En progreso* a *Preparado,* asigna el EGA al auditor que esté en la siguiente posición en la cadena de asignación, cambia el estado de su ITGC concreto de *En evaluación* a *Sin excepciones* o *Con excepciones,* en función de lo seleccionado, y guarda el campo *Justificación* en el ITGC concreto si el usuario lo ha introducido.

Otras funcionalidades

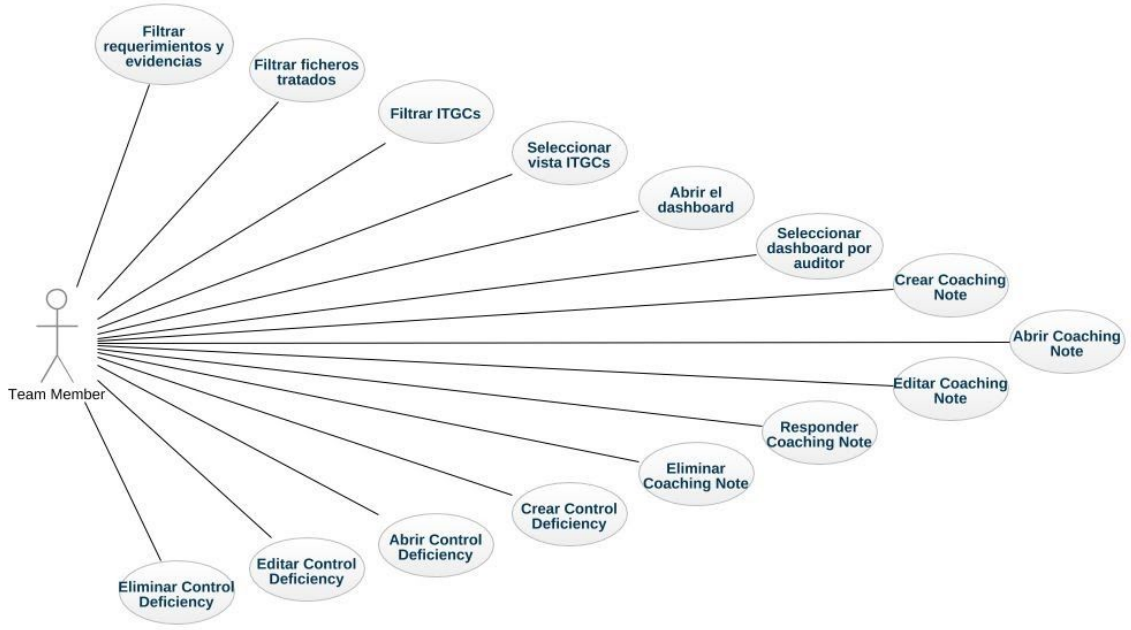


Figura 36. Diagrama de casos de uso para las otras funcionalidades del sistema

28.- Filtrar por condiciones

- **CU-81 - Filtrar requerimientos y evidencias:**
  - El usuario selecciona la opción “Requerimientos de información” en el menú principal de la base.
  - El sistema abre una nueva vista con el listado de todos los requerimientos de información de la base agrupados por su área de aplicación.
  - El usuario selecciona la opción “Filtros”.
  - El sistema muestra los siguientes campos que puede utilizar el usuario para filtrar los requerimientos: listado con las áreas de aplicación del sistema, campo de texto vacío para el nombre del requerimiento y listado de ITGCs de la base asignados a algún requerimiento.
  - El usuario selecciona una área de aplicación del listado, introduce un texto como nombre y/o selecciona el ITGC asociado al requerimiento del listado y una vez finalizado selecciona la opción “Filtrar”.
  - El sistema muestra el listado de requerimientos que corresponden al área seleccionada, que contienen el texto introducido en su nombre y están asociados al control seleccionado.
  - Si el usuario selecciona uno de estos requerimientos, el sistema desplegará el listado de evidencias de ese requerimiento.

- **CU-82 - Filtrar ficheros tratados:**
  - El usuario selecciona la opción “Ficheros” en el menú principal principal de la base.
  - El sistema abre una ventana con el listado de todos los ficheros tratados de la base agrupados por su área de aplicación.
  - El usuario selecciona la opción “Filtros”.
  - El sistema muestra los siguientes campos que puede utilizar el usuario para filtrar los requerimientos: listado con las áreas de aplicación del sistema y campo de texto vacío para el nombre del fichero.
  - El usuario selecciona el área de aplicación del listado e introduce un texto como nombre y una vez finalizado selecciona la opción “Filtrar”.
  - El sistema muestra el listado de requerimientos que corresponden al área seleccionada y que contienen el texto introducido en el nombre del fichero en su nombre.

- **CU-83 - Filtrar ITGCs:**
  - El usuario selecciona la opción “ITGCs” en el menú principal de la base.
  - El sistema abre una vista con el listado con todos los ITGCs concretos de la base.
  - El usuario selecciona la opción “Filtros”.
  - El sistema muestra los siguientes campos que puede utilizar el usuario para filtrar los requerimientos: listado con las áreas de aplicación del sistema, tipo de ITGC (específico de sistema o transversal), listado con los sistemas del cliente (solo para ITGCs específicos de sistema), estado del control, listado de ITGCs genéricos y un campo de texto vacío para el nombre del ITGC.
  - El usuario selecciona los campos de los listados proporcionados que considere y opcionalmente introduce el nombre del ITGC, una vez finalizado selecciona la opción “Filtrar”.
  - El sistema muestra el listado de ITGCs que cumplen con las características seleccionadas y que contienen el texto introducido en el nombre del ITGC en su nombre.

- **CU-84 - Seleccionar vista ITGCs:**
  - El usuario selecciona la opción “ITGCs” en el menú principal de la base.
  - El sistema abre una nueva vista con el listado con todos los ITGCs concretos de la base.
  - El usuario selecciona entre las dos vistas distintas para los ITGCs, consistentes en “Agrupación por sistema” o “Agrupación por ITGC genérico”.
  - El sistema ordena y agrupa los elementos en función de la opción seleccionada.

  La opción “Agrupación por sistema” agrupa los elementos en el siguiente orden: sistema, área de aplicación e ITGC genérico y la opción “Agrupación por ITGC genérico” agrupa los elementos en el siguiente orden: área de aplicación, ITGC genérico y sistema. La vista predeterminada será la de “Agrupación por ITGC genérico”.

29.- Abrir el dashboard

- **CU-85 - Abrir dashboard:**
  - El usuario selecciona la opción “Dashboard” en el menú principal de la base
  - El sistema abre una nueva vista con varios listados: el primer listado consiste en todos los EGAs asignados actualmente al usuario, el segundo listado consiste en todos los EGAs que el usuario ya ha preparado, y que por lo tanto se encuentran asignados a otro auditor, y el tercer listado corresponde a los EGAs en los que el usuario se encuentra designado como revisor pero que aún no están en estado *Preparado.*
  - El sistema muestra para cada uno de los EGAs del listado el auditor actual asignado, el estado del EGA, el número de evidencias y ficheros tratados enlazados en su estructura y el número de *Coaching Notes* asociadas al EGA.
  - Si el usuario selecciona el número de evidencias y ficheros:
    - El sistema abre una ventana con el listado de evidencias y ficheros a los que se referencia.
    - Si el usuario selecciona un elemento y luego selecciona la opción “Abrir”:
      - El sistema abre la evidencia o el fichero seleccionado.
  - Si el usuario selecciona el número de *Coaching Notes:*
    - El sistema abre una nueva ventana con el listado de los títulos de las CNs asociadas al EGA.
    - Si el usuario selecciona una CN y luego selecciona la opción “Abrir”:
      - El sistema abre la CN seleccionada.

- **CU-86 - Seleccionar dashboard por auditor:**
  - El usuario selecciona la opción “Dashboard” en el menú principal de la base.
  - El sistema abre la vista del “Dashboard” y en esta vista muestra también un campo con el nombre del auditor, que por defecto corresponde siempre al usuario debido a que el dashboard abierto también le corresponde.
  - El usuario selecciona este campo y el sistema despliega un listado con todos los auditores de la base.
  - El usuario selecciona un auditor del listado y selecciona la opción “Filtrar”.
  - El sistema le muestra la vista del “Dashboard” correspondiente a ese auditor.

30.- Administración de “Coaching Notes”

- **CU-87 - Crear coaching note:**
  - El usuario selecciona un EGA de la base y luego selecciona la opción “Crear Coaching Note”.
  - El sistema abre una nueva ventana en la que el usuario introduce en el campo “Título” el título de la CN, selecciona de un listado con todos los auditores el auditor al que estará asignada (puede ser él mismo) y opcionalmente introduce en el campo “Descripción” los detalles más concretos de esa CN particular.
  - Una vez finalizado el usuario selecciona la opción “Crear”.
  - En el caso de que no exista otra CN para el mismo EGA con el mismo título, el sistema creará la CN en la base con los parámetros introducidos
  - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-88 - Abrir Coaching Note:**
  - El usuario selecciona un EGA de la base y luego selecciona las *Coaching Notes* asociadas a éste.
  - El sistema abre una nueva ventana en la que se muestra el listado con los títulos de las CNs asociadas a ese EGA.
  - El usuario selecciona una de las CNs y luego selecciona la opción “Abrir”.
  - El sistema abre una nueva ventana en la que se muestra el título de la CN, su descripción, el usuario asignado a esta y su respuesta (en caso de que exista).

- **CU-89 - Editar Coaching Note:**
  - El usuario abre una CN asociada a un EGA de la base.
  - En el caso de que el usuario sea el creador de la CN:
    - El sistema muestra la opción “Editar” y el usuario selecciona esta opción.
    - El sistema le permite modificar los campos mostrados al abrir la CN.
    - El usuario modifica los campos deseados y una vez finalizado selecciona la opción “Guardar”.
    - En caso de que no exista otra CN para el mismo EGA con el mismo título el sistema guardará los cambios realizados,
    - En el caso contrario el sistema muestra un mensaje informando del error.

- **CU-90 - Responder coaching note:**
  - El usuario abre una CN asociada a un EGA de la base.
  - En caso de que el usuario esté asignado a esa CN:
    - El sistema muestra la opción “Responder” y el usuario selecciona esta opción.
    - El sistema le permite modificar el campo “Respuesta” y el usuario introduce o modifica el campo “Respuesta”.
    - Una vez finalizado el usuario selecciona la opción “Guardar”.
    - El sistema guarda el cambio realizado en el campo respuesta.

- **CU-91 - Eliminar coaching note:**
  - El usuario selecciona un EGA de la base y luego selecciona las *Coaching Notes* asociadas a éste.
  - El sistema abre una nueva ventana en la que se muestra un listado con los títulos de las CNs asociadas a ese EGA.
  - El usuario selecciona una de las CNs.
  - En caso de que sea el creador de esta o tenga el rol *Team leader:*
    - El sistema muestra la la opción “Eliminar” y el usuario selecciona esta opción.
    - El sistema abre una nueva ventana solicitando la confirmación de la acción y el usuario selecciona la opción “Eliminar”.
    - El sistema elimina la CN asociada a ese EGA de la base.

31.- Administración de Control Deficiencies

- **CU-92 - Crear Control Deficiency:**
  - El usuario selecciona un ITGC del sistema y luego selecciona la opción “Crear Control Deficiency” o al marcar un EGA como preparado selecciona la opción “Sí” cuando el sistema pregunta para crear una *Control Deficiency.*
  - El sistema abre una nueva ventana en la que el usuario introduce la descripción de la deficiencia identificada, la gravedad de la deficiencia, selecciona si existe o no un control mitigante y en caso que la selección sea afirmativa introduce la descripción del control mitigante.
  - Una vez finalizado selecciona la opción “Guardar”.
  - El sistema guarda la *Control Deficiency* asociada al ITGC.
  - En el caso contrario el sistema muestra un mensaje informando de esto.

- **CU-93 - Abrir Control Deficiency:**
  - El usuario selecciona un ITGC del sistema, selecciona su *Control Deficiency* (en caso de que exista), y luego selecciona la opción “Abrir *Control Deficiency”.*
  - El sistema abre una nueva ventana en la que se muestra la descripción de la deficiencia detectada en el ITGC, la gravedad de esta deficiencia, si existe algún control mitigante y la descripción de este control.

- **CU-94 - Editar Control Deficiency:**
  - El usuario abre la *Control Deficiency* de un ITGC del sistema.
  - En el caso de que el usuario esté en la cadena de asignación del EGA que documenta ese ITGC o tenga el rol *Team Leader.*
    - EL sistema le muestra la opción “Editar” y el usuario la selecciona.
    - El sistema abre una nueva ventana con la información actual de la *Control Deficiency.*
    - El usuario modifica la descripción, la gravedad de la deficiencia, selecciona si existe un control mitigante y/o modifica la descripción del control mitigante.
    - Una vez finalizado el usuario selecciona la opción “Guardar”.
    - El sistema actualiza el *Control Deficiency* con los campos introducidos por el usuario.

- **CU-95 - Eliminar Control Deficiency:**
  - El usuario selecciona un ITGC del sistema y selecciona su *Control Deficiency* (en caso de que exista).
  - En el caso de que el usuario esté en la cadena de asignación del EGA que documenta ese ITGC o tenga el rol *Team Leader.*
    - El sistema muestra la opción “Eliminar*”* y el usuario la selecciona.
    - El sistema abre una nueva ventana en la que solicita la confirmación de la acción y el usuario selecciona la opción “Eliminar”.
    - El sistema elimina la deficiencia asociada a ese ITGC.

Responsable IT Cliente

Los casos de uso asignados de este actor se corresponden a las funcionalidades descritas en el punto *2.1.2 - Funcionalidades principales* para los responsables de IT de los clientes. Las funcionalidades de este actor en el sistema se encuentran muy limitadas y los casos de uso a los que tendrá acceso serán los justos para la correcta realización de estas funcionalidades.

Se han agrupado todos los casos de uso del actor en un sólo diagrama, las funcionalidades a las que dan respuesta estos casos de uso son las siguientes: *32.- Acceso al sistema, 33.- Consultar requerimientos de información, 34.- Añadir y eliminar evidencias* y *35.- Recibir recordatorio.*

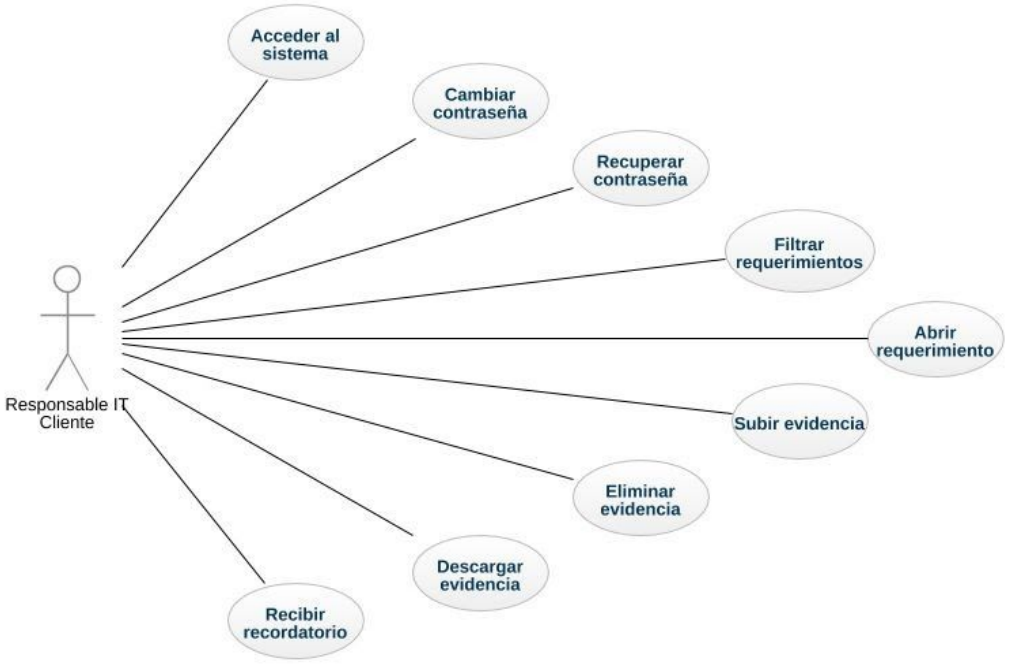


Figura 37. Diagrama de casos de uso para las funcionalidades de los responsables de IT de los clientes

32.- Acceso al sistema

- **CU-96 - Acceder al sistema:**
  - El usuario selecciona el link de acceso al sistema, el cual se encuentra en los mails enviados por el sistema al cliente y en su navegador compatible se abre una nueva ventana con la interfaz del sistema para los clientes.
  - El sistema muestra los campos “Email” y “Contraseña”.
  - El usuario introduce su dirección de correo electrónico corporativa y su contraseña registrada y una vez introducidos selecciona la opción “Log In”.
  - En el caso de que el correo electrónico esté registrado en el sistema y la contraseña coincida, el sistema identifica al usuario y abre una nueva vista con el listado de los requerimientos de información asignados a él.
  - En el caso contrario, el sistema muestra un mensaje informando de que el email o la contraseña no son válidos.

- **CU-97 - Cambiar contraseña:**
  - El usuario se autentifica correctamente en el sistema y luego selecciona la opción “Cambiar Contraseña”.
  - El sistema abre una nueva ventana en la que el usuario introduce su contraseña actual, luego introduce dos veces la nueva contraseña y una vez finalizado selecciona la opción “Cambiar”.
  - En el caso de que la contraseña actual introducida sea incorrecta, o si las nuevas contraseñas no coinciden o no cumplen con la complejidad definida, el sistema muestra un mensaje de error informando al usuario.
  - En el caso de que todo esté correcto, el sistema actualiza la contraseña del usuario con la nueva contraseña introducida.

- **CU-98 - Recuperar contraseña:**
  - El usuario abre la interfaz para los clientes del sistema y luego selecciona la opción “Recuperar contraseña”.
  - El sistema envía a la dirección de email del usuario un correo con un enlace temporal para cambiar la contraseña.
  - El usuario selecciona este enlace y se abre una nueva ventana del sistema en la que el usuario introduce su nueva contraseña dos veces.
  - En el caso de que las contraseñas coincidan y cumplan con los parámetros de complejidad establecidos, el sistema actualiza la contraseña del usuario con la nueva contraseña introducida.
  - En el caso contrario, el sistema muestra un mensaje informando del error.

33.- Consultar requerimientos de información

- **CU-99 - Filtrar requerimientos:**
  - El usuario se autentifica en el sistema y este muestra una vista con todos los requerimientos de información pendientes asignados al usuario.
  - El sistema muestra el campo “Estado”, en el que se listan los 3 estados posibles de un requerimiento y el campo “Usuario”, en el que se listan los responsables de IT de ese cliente concreto y la opción “Todos”.
  - El usuario selecciona los estados deseados del listado “Estado” y/o los usuarios deseados del listado “Usuario” y luego selecciona la opción “Filtrar”.
  - El sistema muestra los requerimientos cuyo estado corresponde con alguno de los seleccionados y su usuario asignado a alguno de los seleccionados.

- **CU-100 - Abrir requerimiento:**
  - El usuario selecciona uno de los requerimientos de la base y luego selecciona la opción “Abrir”.
  - El sistema abre una nueva ventana con el nombre del requerimiento, su estado, la descripción, la información requerida, la aclaración (en caso de que exista) y las evidencias subidas para ese requerimiento (en caso de que exista alguna).
  - Si el requerimiento está asignado al usuario y en estado *Pendiente de recibir*:
    - El sistema muestra al usuario la opción “Añadir evidencia” y al seleccionar una evidencia muestra la opción “Eliminar”.

34.- Añadir y eliminar evidencias

- **CU-101 - Añadir evidencia:**
  - El usuario abre un requerimiento de la base en estado *Pendiente de recibir* y asignado a él.
  - El sistema despliega un listado con todas las evidencias subidas para ese requerimiento y luego el usuario selecciona la opción “Añadir evidencia”.
  - El sistema abre una nueva ventana en la que el usuario introduce el nombre de la evidencia y selecciona de su dispositivo el fichero a subir que contiene esta evidencia.
  - El usuario selecciona la opción “Guardar”.
  - En el caso que no exista una evidencia con el mismo nombre para ese requerimiento, el sistema almacena la nueva evidencia subida, y en el caso contrario muestra un mensaje informando del error.

- **CU-102 - Eliminar evidencia:**
  - El usuario abre un requerimiento de la base en estado *Pendiente de recibir* y asignado a él.
  - el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia del listado y luego selecciona la opción “Eliminar”.
  - El sistema muestra un mensaje solicitando la confirmación del usuario. e
  - El usuario selecciona la opción “Eliminar” y la evidencia seleccionada queda eliminada del sistema.

- **CU-103 - Descargar evidencia:**
  - El usuario abre un requerimiento y el sistema despliega un listado con todas las evidencias subidas para ese requerimiento.
  - El usuario selecciona una evidencia y luego selecciona la opción “Descargar”.
  - El sistema descarga el fichero que contiene la evidencia seleccionada en el dispositivo del usuario.

35.- Recibir recordatorios

- **CU-104 - Recibir recordatorio:**
  - El usuario recibe un mensaje en su dirección de correo electrónico enviado por el sistema en el que se le recuerdan los requerimientos que tiene asignados y en estado pendiente. En ese mismo mail se encuentra un link al sistema para que el usuario lo seleccione y acceda directamente.

4.- Esquema de la base de datos del sistema en pseudocódigo

Oficina {
    id_oficina: *Integer*
    ciudad: *String* {*Not null*}
    pais: *String* {*Not null*}
    Constraints: *Unique* {ciudad, pais}
}

Cliente {
    id_cliente: *Integer*
    nombre: *String* {*Not null*}
    descripcion: *String*
    oficina: *Integer* {*Not null*}
    ForeignKeys*:** {oficina} ***References Oficina {id_oficina}
}

Auditor {
    id_auditor: *Integer*
    username: *String* {*Not null, Unique*}
    nombre: *String* {*Not null*}
    perfil: *enum*('Manager', 'Senior', 'Junior') {*Not null*}
    mail: *String* {*Not null, Unique*}
    oficina: *Integer* {*Not null*}
    ForeignKeys: {oficina} ***References*** Oficina {id_oficina}
}

Manager {
    id_auditor: *Integer*
    ForeignKeys*:** {id_usuario} ***References Auditor {id_auditor}
}

Responsable_IT {
    id_responsable: *Integer*
    nombre: *String* {*Not null*}
    mail: *String* {*Not null, Unique*}
    password: *String* {*Not null*}
    descripcion: *String*
    cliente: *Integer* {*Not null*}
    ForeignKeys*:** {cliente} ***References Cliente {id_cliente}
}

Sistema_estandar {
    id_sistema_stdr: *Integer*

nombre: *String* {*Not null*, *Unique*}
tipo_sistema: *enum*('Aplicacion', 'Base de datos', 'Sistema operativo') {*Not null*}
descripcion: *String*
creador: Integer {*Not null*}
ForeignKeys*:** {creador} ***References Manager {id_auditor}
}

Sistema_concreto {
id_sistema: *Integer*
nombre: *String* {*Not null*}
descripcion: *String*
tipo_sistema: *enum*('Aplicacion', 'Base de datos', 'Sistema operativo')
en_alcance: *boolean* {*Not null*}
cliente: *Integer* {*Not null*}
sistema_estandar: *Integer*
ForeignKeys: {cliente} ***References*** Cliente {id_cliente}
{sistema_estandar} ***References*** Sistema_estandar {id_sistema_stdr}
}

ITGC_estandar {
id_ITGC_stdr: *Integer*
nombre: String {*Not null, Unique*}
tipo_control: *enum*('Transversal', 'Específico') {*Not null*}
descripcion: *String*
area: *enum*('Entendimiento', 'Seguridad', 'Gestión de cambios',
'Operaciones', 'Desarrollo') {*Not null*}
creador: *Integer* {*Not null*}
ForeignKeys: {creador} ***References*** Manager {id_auditor}
}

Base {
id_base: *Integer*
activa: *Boolean* {*Not null*}
periodo_fiscal: *Year* {*Not null*}
cliente: *Integer* {*Not null*}
propietario: *Integer* {*Not null*}
ForeignKeys*:** {cliente} ***References Cliente {id_cliente}
{propietario} ***References*** Manager {id_auditor}
***Constraints*:** *Unique* {cliente, periodo_fiscal}
}

Relacion_ITGC_Base {
base: *Integer*

itgc_estandar: *Integer*
ForeignKeys: {base} ***References*** Base {id_base},
{itgc_estandar} ***References*** ITGC_estandar {id_ITGC_stdr}
}

Plantilla_EGA {
id_plantilla: *Integer*
nombre: *String* {*Not null*}
URl_estructura: *String* {*Not null*}
oficina: *Integer* {*Not null*}
creador: *Integer* {*Not null*}
itgc_representado: *Integer* {*Not null*}
ForeignKeys: {oficina} ***References*** Oficina {id_oficina}
{creador} ***References*** Manager {id_auditor}
{itgc_representado} ***References*** ITGC_estandar {id_itgc_stdr}
Constraints: *Unique* {nombre, oficina}
}

Plantilla_especifica_sistema {
id_plantilla: *Integer*
sistema_estandar: *Integer*
ForeignKeys: {id_plantilla} ***References*** Plantilla_EGA {id_plantilla}
{sistema_estandar} ***References*** Sistema_estandar {id_sistema_stdr}
}

Grupo_requerimiento {
id_grupo_req: *Integer*
nombre: *String* {*Not null*}
area: *enum*('Entendimiento', 'Seguridad', 'Gestión de cambios',
'Operaciones','Desarrollo') {*Not null*}
creador: *Integer* {Not null}
ForeignKeys: {creador} ***References*** Manager {id_auditor}
}

Rol {
auditor: *Integer*
base: *Integer*
ForeignKeys: {auditor} ***References*** Auditor {id_auditor}
{base} ***References*** Base {id_base}
}

Req_informacion_estandar {
id_req_stdr: *Integer*
nombre: *String* {*Not null*}
descripcion: *String* {*Not null*}
informacion_requerida: *String* {*Not null*}
creador: *Integer* {*Not null*}
oficina: *Integer* {*Not null*}
grupo: *Integer* {*Not null*}
ForeignKeys*:** {creador} ***References Manager {id_auditor}
{oficina} ***References*** Oficina {id_oficina}
{grupo} ***References*** Grupo_requerimiento {id_grupo_req}
}

Req_estandar_sistema {
id_req_stdr: *Integer*
sistema_especifico: *Integer* {*Not null*}
ForeignKeys: {id_req_stdr} ***References*** Req_informacion_estandar {id_req_stdr}
{sistema_especifico} ***References*** Sistema_estandar {id_sistema_stdr}
}

Relacion_ReqEstandar_ITGCestandar {
req_stdr: *Integer*
ITGC_stdr: *Integer*
ForeignKeys: {req_stdr} ***References*** Req_informacion_estandar {id_req_stdr}
{ITGC_stdr} ***References*** ITGC_estandar {id_ITGC_stdr}
}

Req_informacion {
id_req: *Integer*
identificador: *String* {*Not null*}
nombre: *String* {*Not null*}
descripcion: *String* {*Not null*}
informacion_requerida: *String* {*Not null*}
aclaracion: *String*
estado_req: *enum*('Pendiente', 'En evaluación', 'Recibido') {*Not null*}
grupo: *Integer* {*Not null*}
basado_en: *Integer*
base: *Integer* {*Not null*}
ForeignKeys: {grupo} ***References*** Grupo_requerimiento {id_grupo_req}
{basado_en} ***References*** Req_informacion_estandar {id_req_stdr}
{base} ***References*** Base {id_base}
Constraints: *Unique* {identificador, base}
}

Relacion_ReqInformacion_ResponsableIT {
req_informacion: *Integer*
responsable_IT: *Integer*
ForeignKeys: {req_informacion} ***References*** Req_informacion {id_req}
{responsable_IT} ***References*** Responsable_IT {id_responsable}
}

Evidencia {
id_evidencia: *Integer*
nombre: *String* {*Not null*}
URL_fichero_asociado: *String* {*Not null, Unique*}
req_informacion: *Integer* {*Not null*}
auditor: *Integer*
responsable_IT: *Integer*
ForeignKeys: {req_informacion} ***References*** Req_informacion {id_req}
{auditor} ***References*** Auditor {id_auditor}
{responsable_IT} ***References*** Responsable_IT {id_responsable}
Constraints: *Unique* {nombre, req_informacion}
}

Fichero_tratado {
id_fichero: *Integer*
nombre: *String* {*Not null*}
URL_fichero_asociado: *String* {*Not null, Unique*}
area: *enum*('Entendimiento', 'Seguridad', 'Gestión de cambios',
'Operaciones','Desarrollo') {*Not null*}
base: *Integer* {*Not null*}
auditor: *Integer* {*Not null*}
ForeignKeys: {base} ***References*** Base {id_base}
{auditor} ***References*** Auditor {id_auditor}
}

Politica_recordatorio {
base: *Integer*
siempre: *Boolean* {*Not null*}
periodicidad: *Integer* {*Not null*}
mensaje: *String* {*Not null*}
ForeignKeys: {base} ***References*** Base {id_base}
}

ITGC_concreto {

id_ITGC: *Integer*

nombre: *String* {*Not null*}

descripcion: *String* {*Not null*}

estado: *enum*('En evaluacion', 'Sin excepciones', 'Con excepciones') {*Not null*}

area: *enum*('Entendimiento', 'Seguridad', 'Gestion de cambios',
'Operaciones','Desarrollo') {*Not null*}

justificacion: *String*

itgc_representado: *Integer* {*Not null*}

base: *Integer* {*Not null*}

ForeignKeys: {itgc_representado} ***References*** ITGC_estandar {id_itgc_stdr}
{base} ***References*** Base {id_base}

Constraints**:** *Unique* {nombre, base}

}

ITGC_concreto_de_sistema {

id_ITGC: *Integer*

sistema_concreto: *Integer* {*Not null*}

ForeignKeys: {sistema_concreto} ***References*** Sistema_concreto {id_sistema}

}

Control_Deficiency {

id_CD: *Integer*

nombre: *String* {*Not null*}

descripcion: *String* {*Not null*}

control_mitigante: *Boolean* {*Not null*}

descripcion_mitigante: *String*

gravedad_deficiencia: *enum*('Baja', 'Media', 'Alta') {*Not null*}

itgc_concreto: *Integer* {*Not null*}

ForeignKeys: {itgc_concreto} ***References*** ITGC_concreto {id_ITGC}

Constraints**:** *Unique* {nombre, itgc_concreto}

}

Relacion_ReqInformacion_ITGCconcreto {

req_informacion: *Integer*

itgc_concreto: *Integer*

ForeignKeys: {req_informacion} ***References*** Req_informacion {id_req}
{itgc_concreto} ***References*** ITGC_concreto {id_ITGC}

}

EGA {

id_EGA: *Integer*

itgc_concreto: *Integer* {*Not null, Unique*}

locked: *Boolean* {*Not null*}

URL_documento: *String* {*Not null, Unique*}

estado: *enum*('En progreso', 'Preparado', 'Revisado', 'Cerrado')

asignado_a: *Integer*

accedido_por: *Integer*

plantilla_usada: *Integer*

ForeignKeys: {itgc_concreto} ***References*** ITGC_concreto {id_ITGC}
{asignado_a} ***References*** Auditor {id_auditor}
{accedido_por} ***References*** Auditor {id_auditor}
{plantilla_usada} ***References*** Plantilla {id_plantilla}

}

Cadena_asignacion {

EGA: *Integer*

auditor: *Integer*

posicion: *Integer* {*Not null*}

ForeignKeys: {EGA} ***References*** EGA {id_EGA}
{auditor} ***References*** Auditor {id_auditor}

}

Coaching_Note {

id_CN: *Integer*

EGA: *Integer* {*Not null*}

titulo: *String* {*Not null*}

descripcion: *String*

respuesta: *String*

creada_por: *Integer* {*Not null*}

asignada_a: *Integer* {*Not null*}

ForeignKeys: {EGA} ***References*** EGA {id_EGA}
{creada_por} ***References*** Auditor {id_auditor}
{asignada_a} ***References*** Auditor {id_auditor}

Constraints: *Unique* {EGA, titulo}

}

Relacion_EGA_FicheroTratado {

EGA: *Integer*

fichero_tratado: *Integer*

ForeignKeys: {EGA} ***References*** EGA {id_EGA}
{fichero_tratado} ***References*** Fichero_tratado {id_fichero}

}

Relacion_EGA_Evidencia {

EGA: *Integer*

evidencia: *Integer*

ForeignKeys: {EGA} ***References*** EGA {id_EGA}

{evidencia} ***References*** Fichero_tratado {id_evidencia}

}

Relacion_EGA_EGA {

EGA_principal: *Integer*

EGA_enlazado: *Integer*

ForeignKeys: {EGA_principal} ***References*** EGA {id_EGA}

{EGA_enlazado} ***References*** EGA {id_EGA}

}